船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

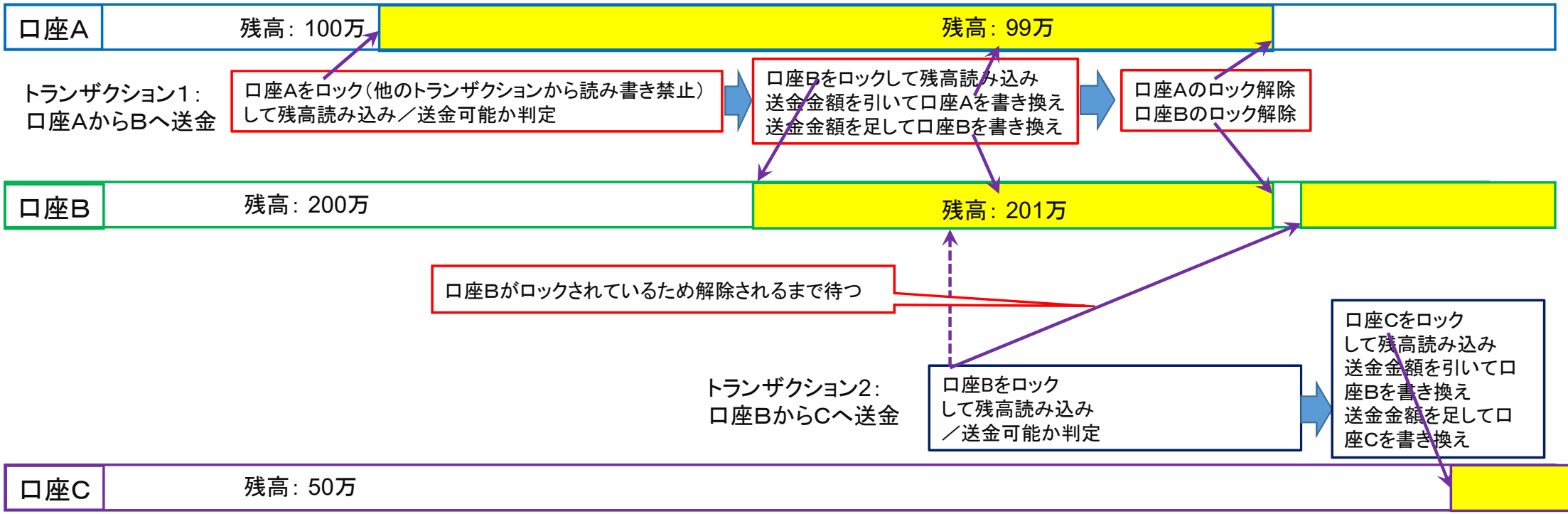
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

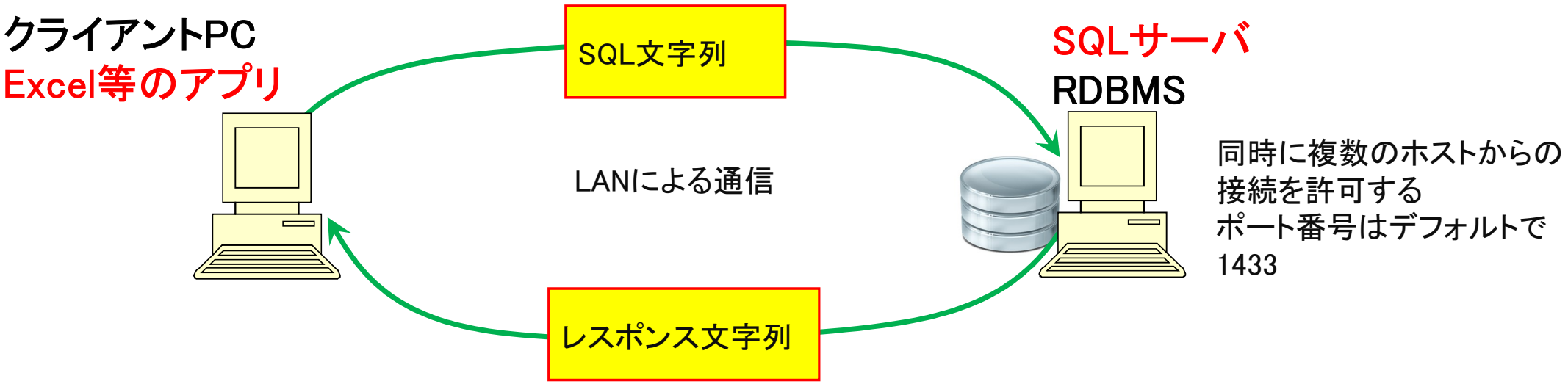
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

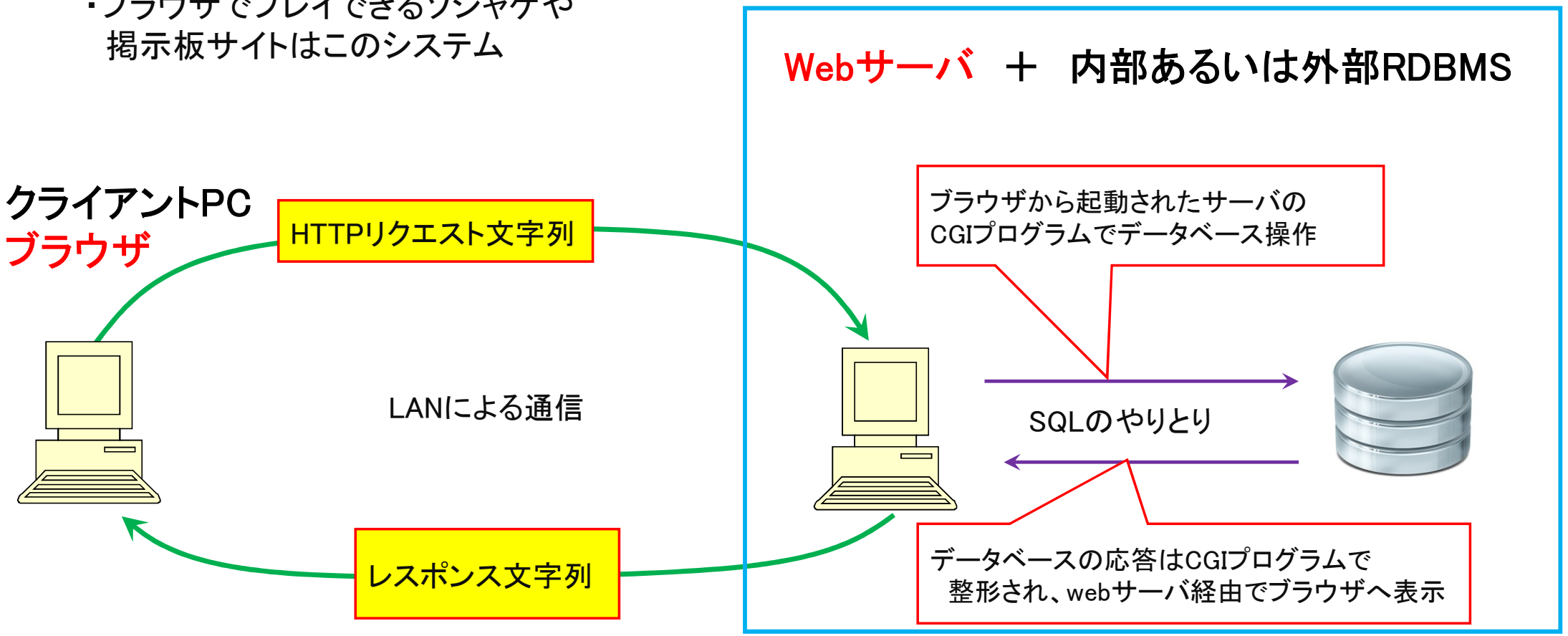
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

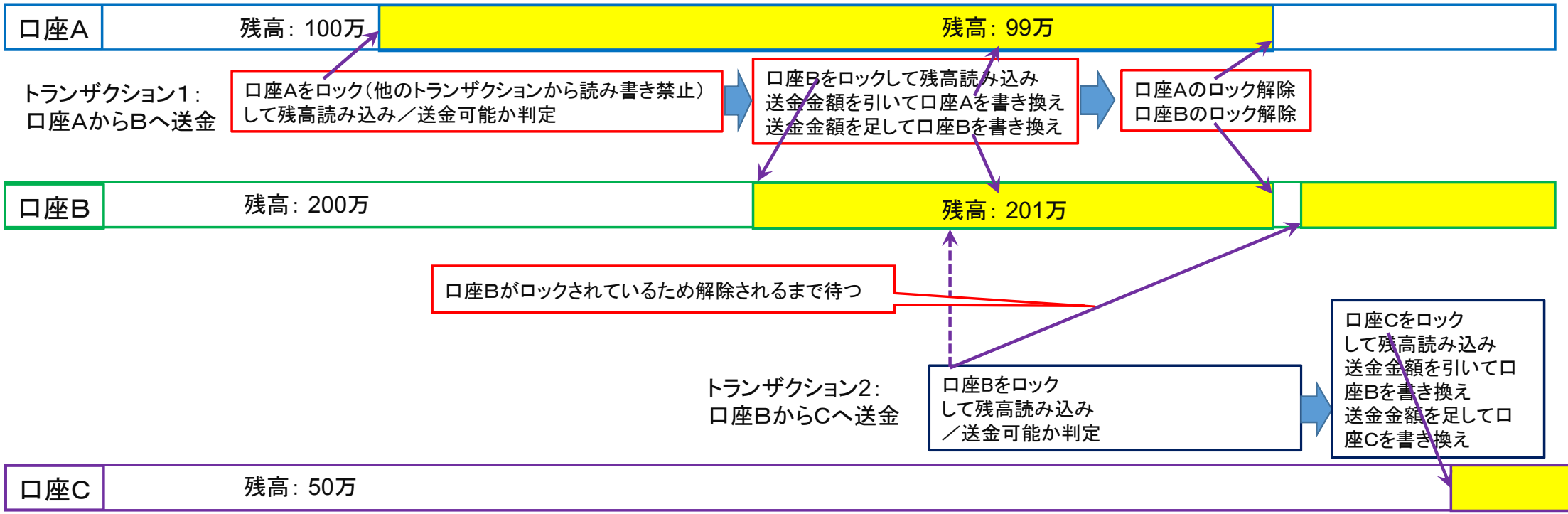
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

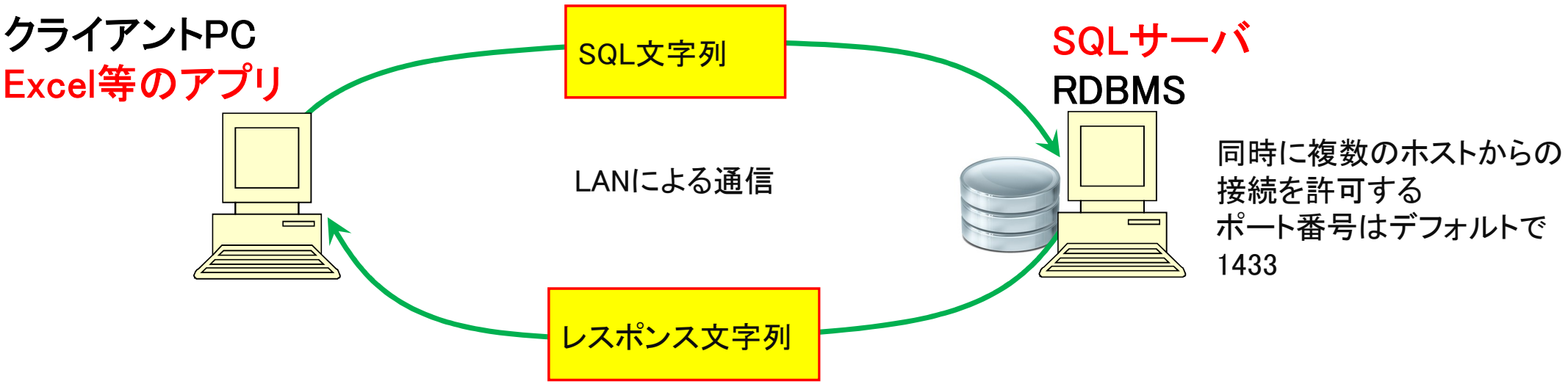
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

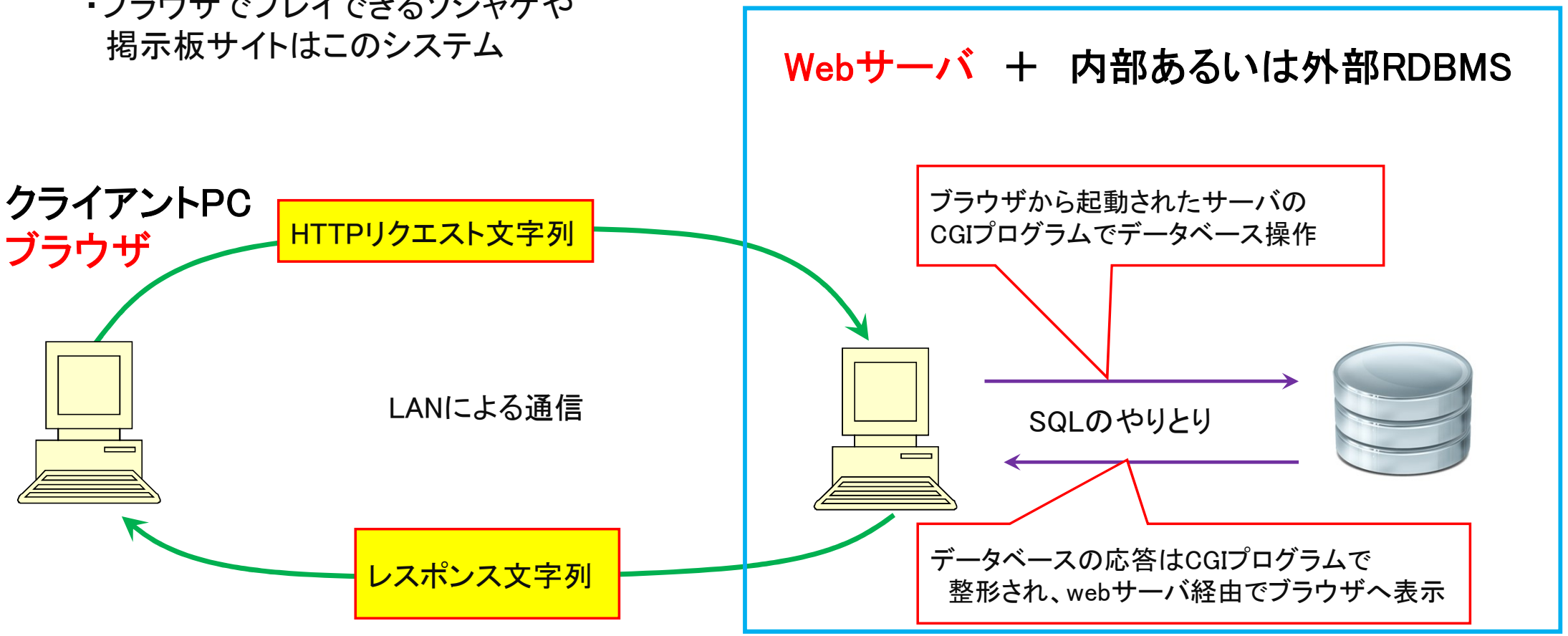
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

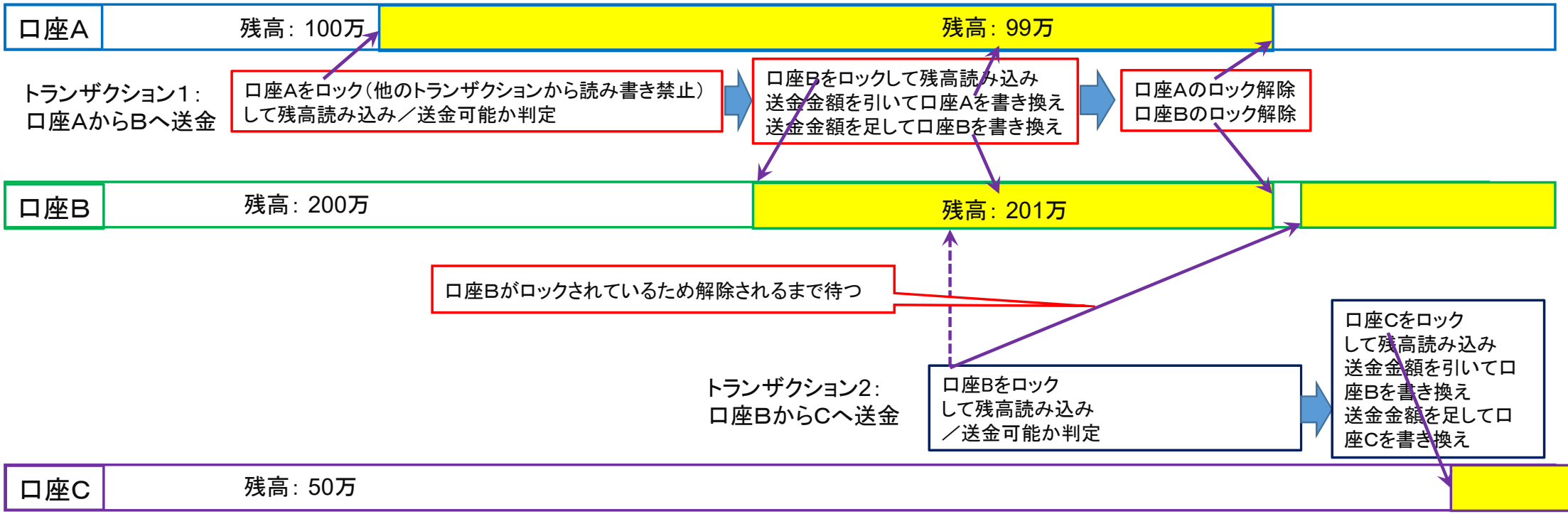
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

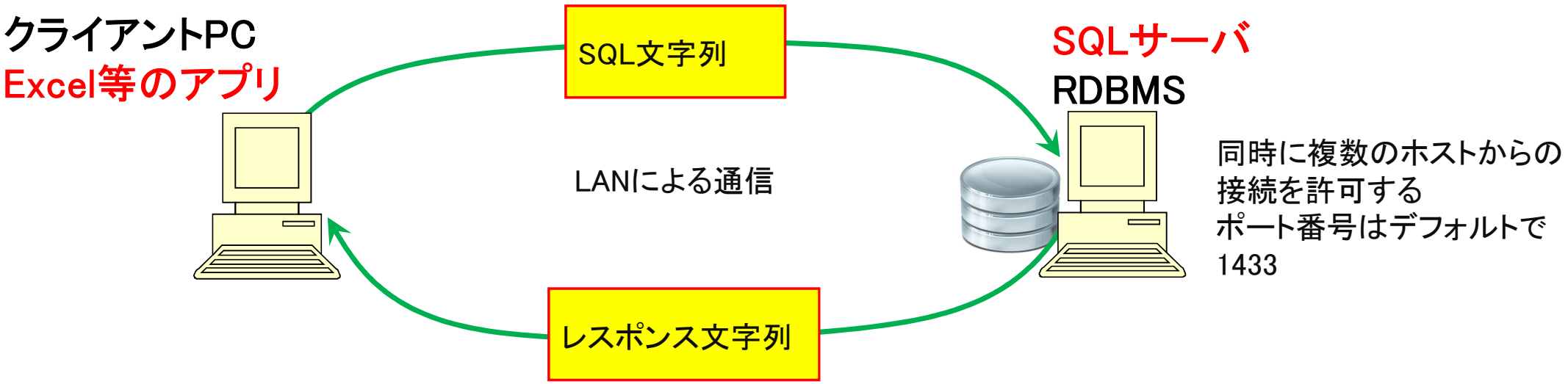
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

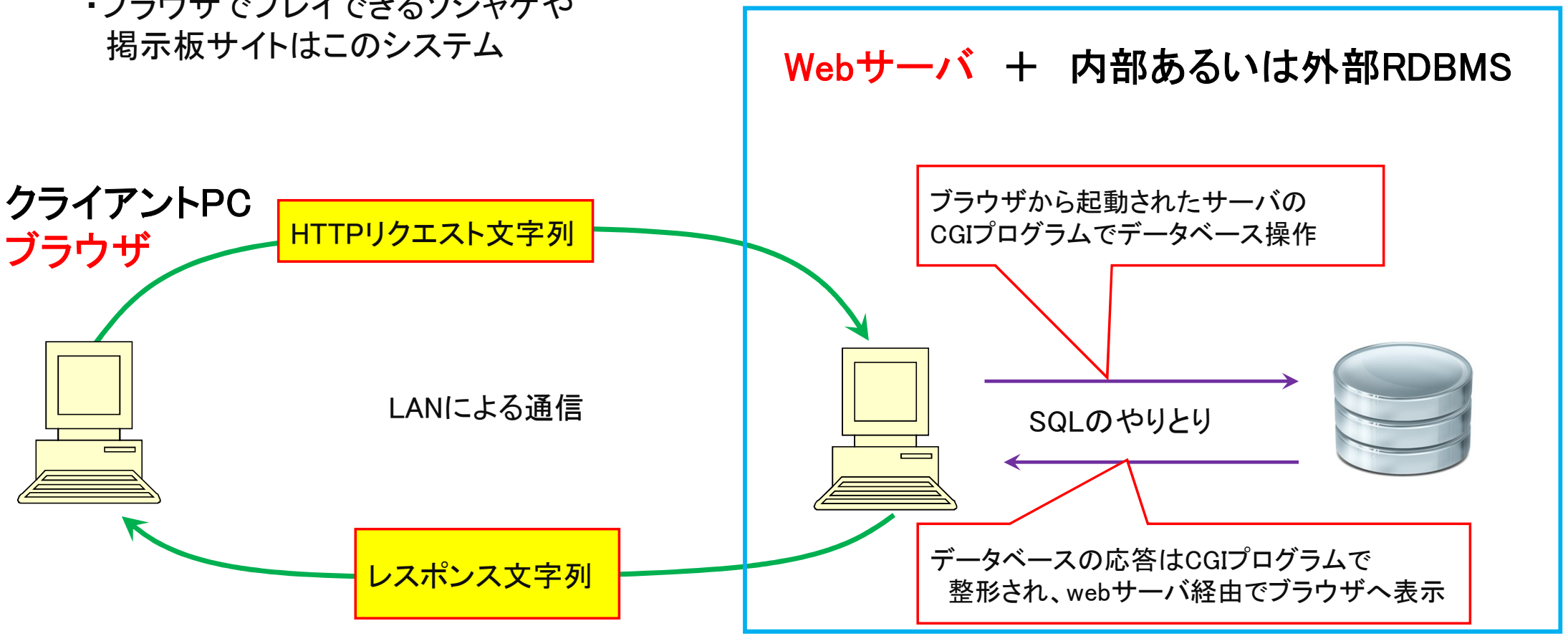
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

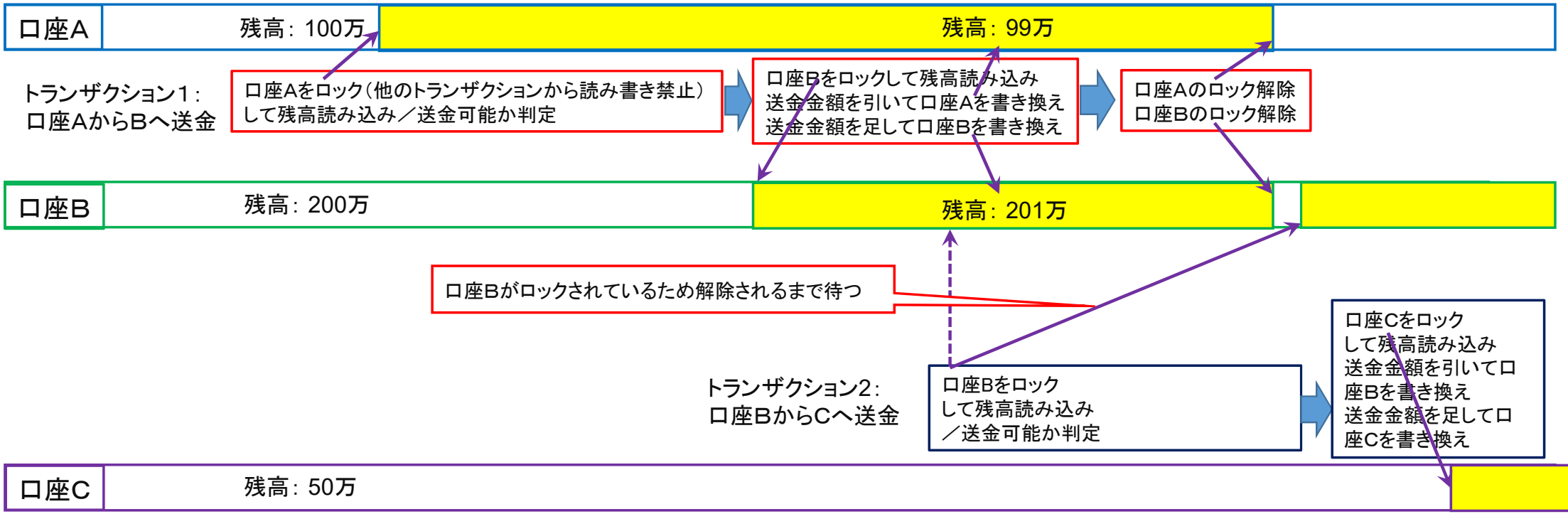
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

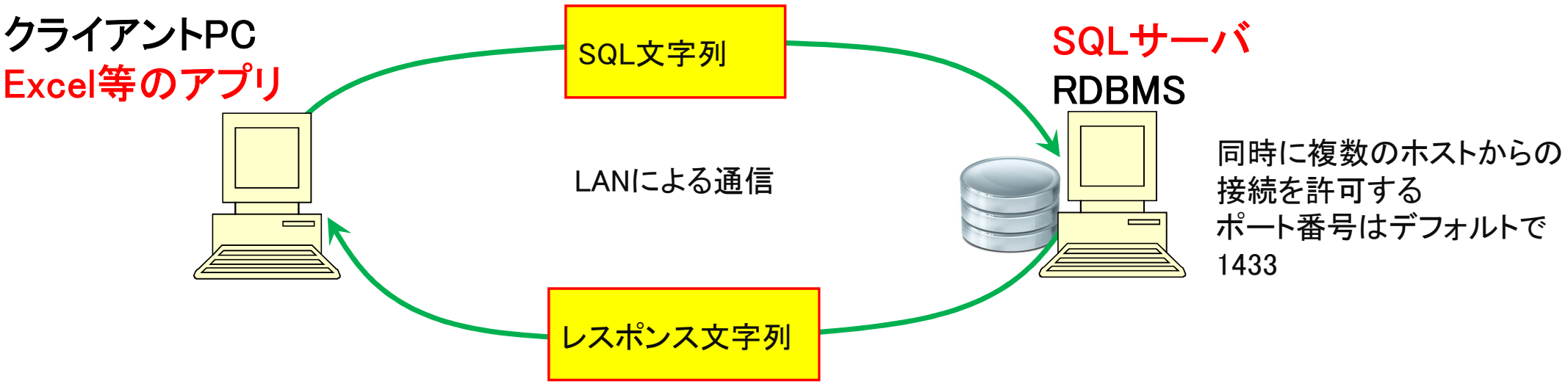
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

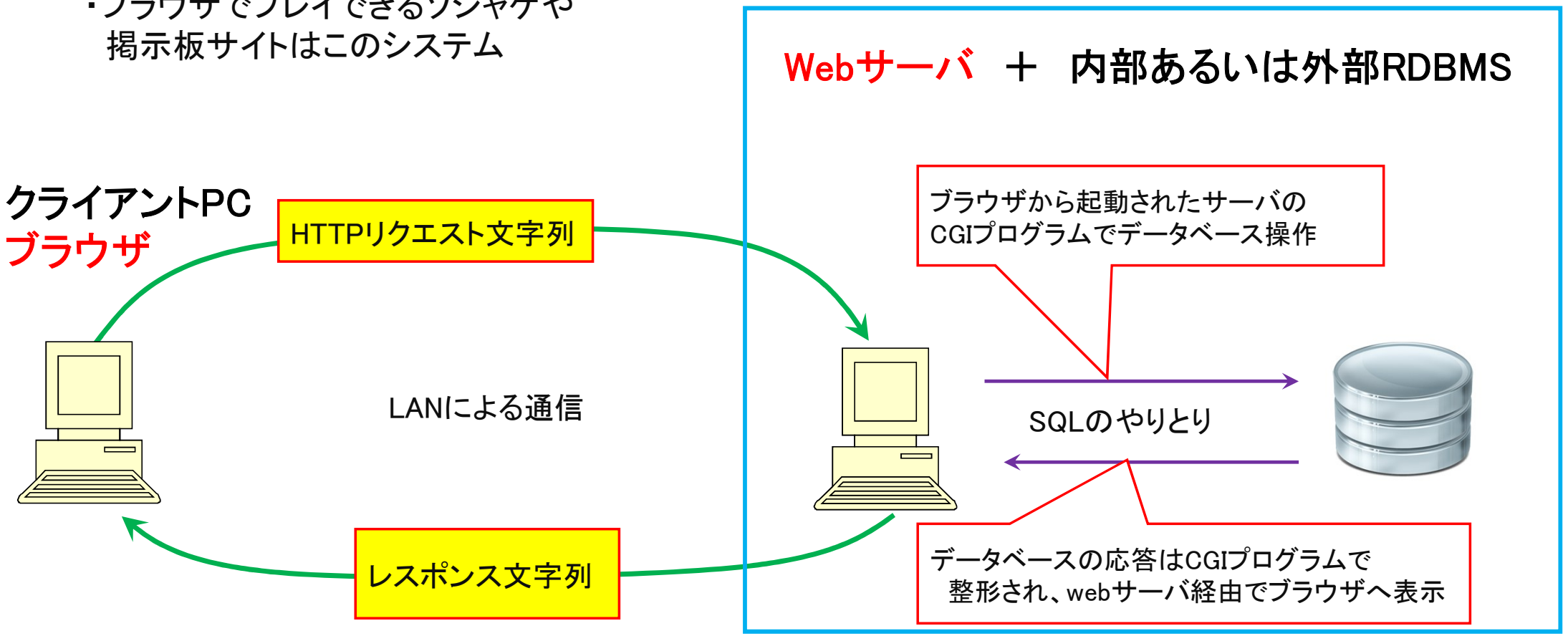
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

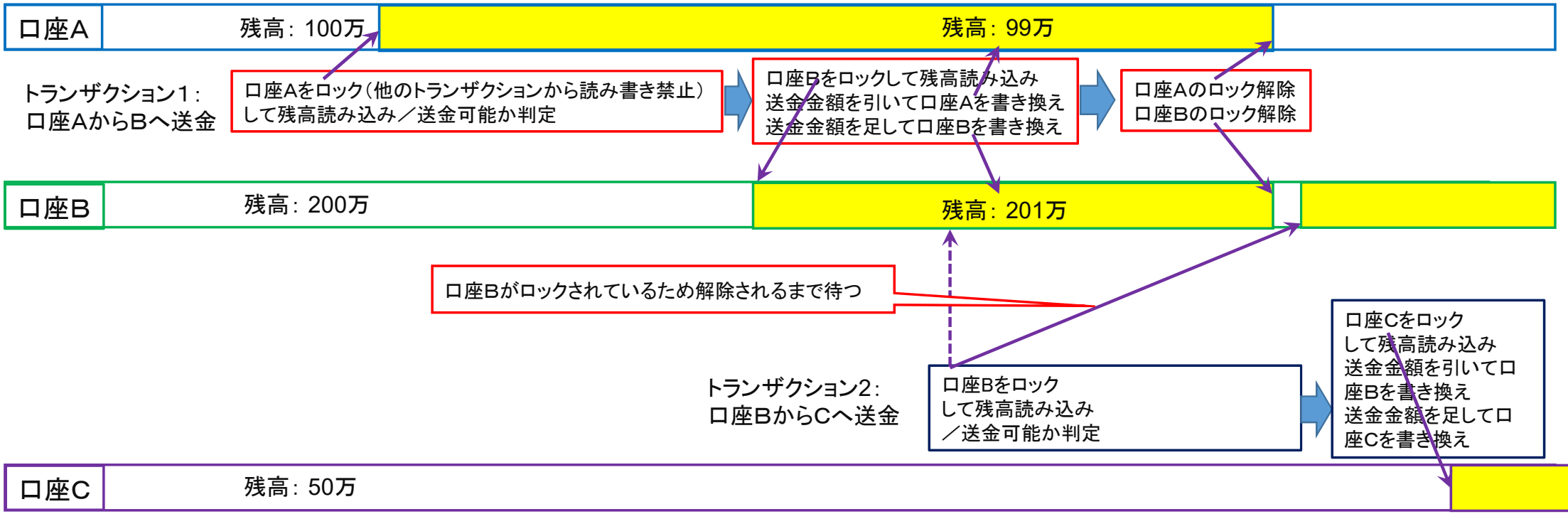
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

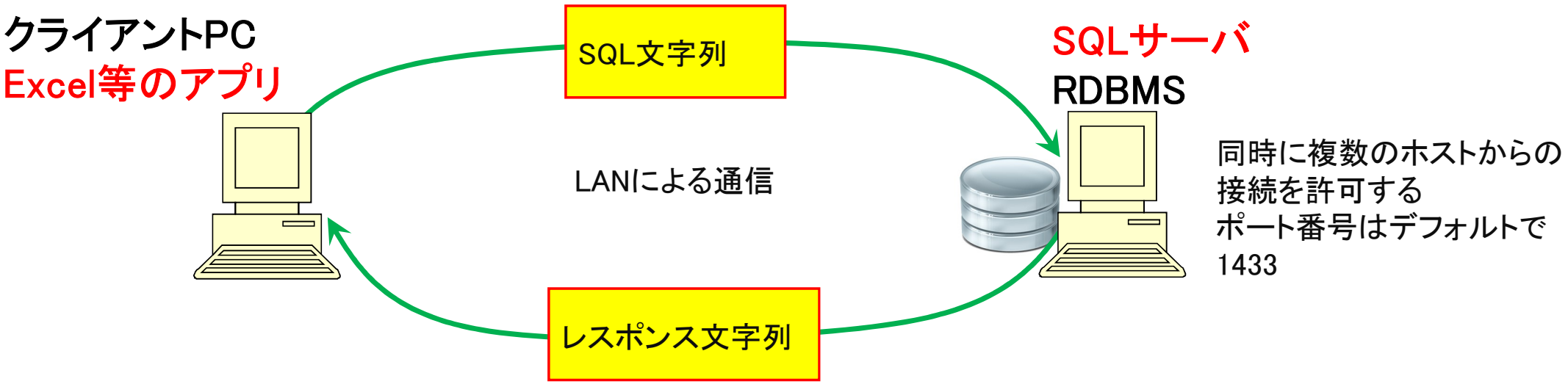
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

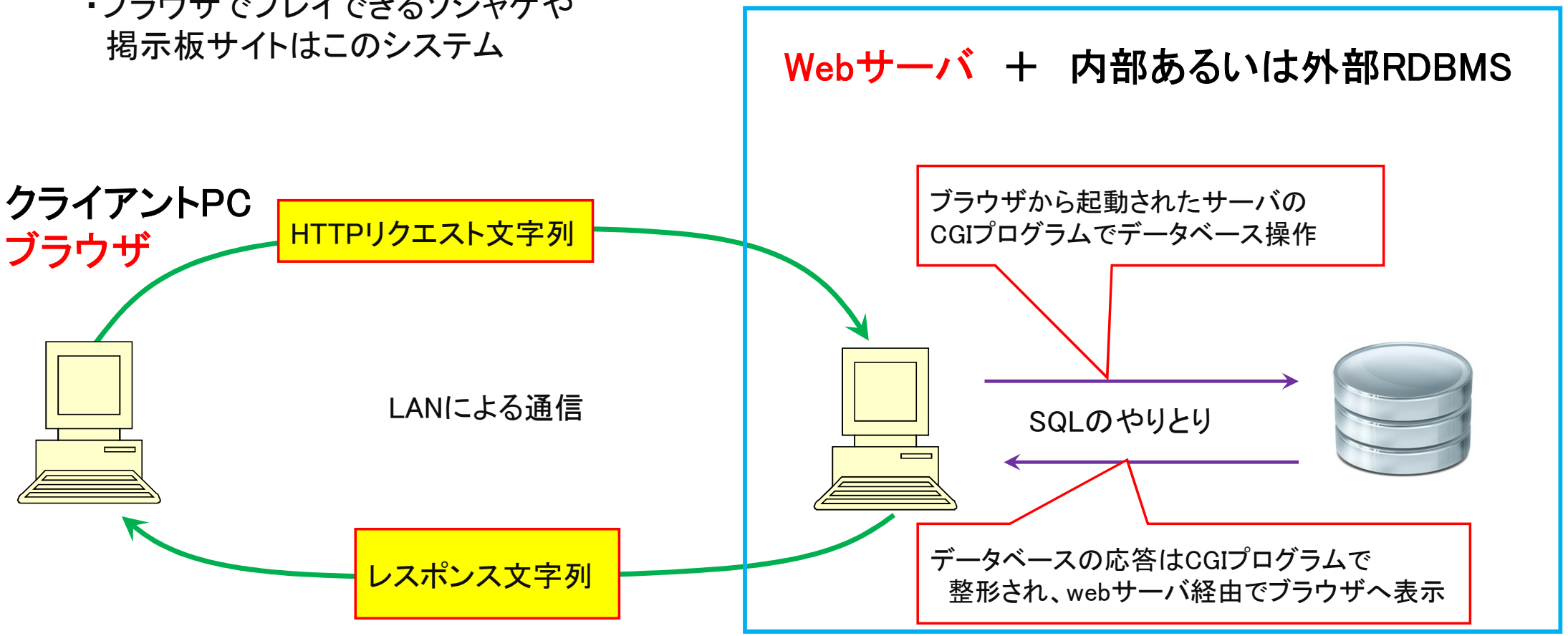
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

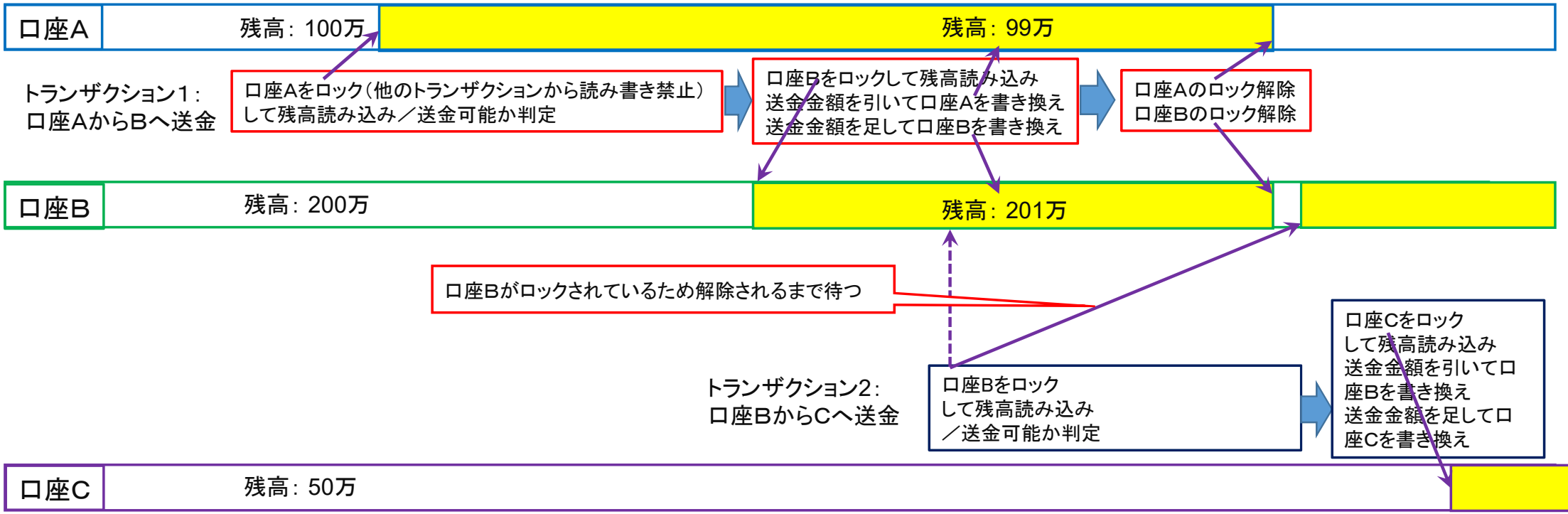
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

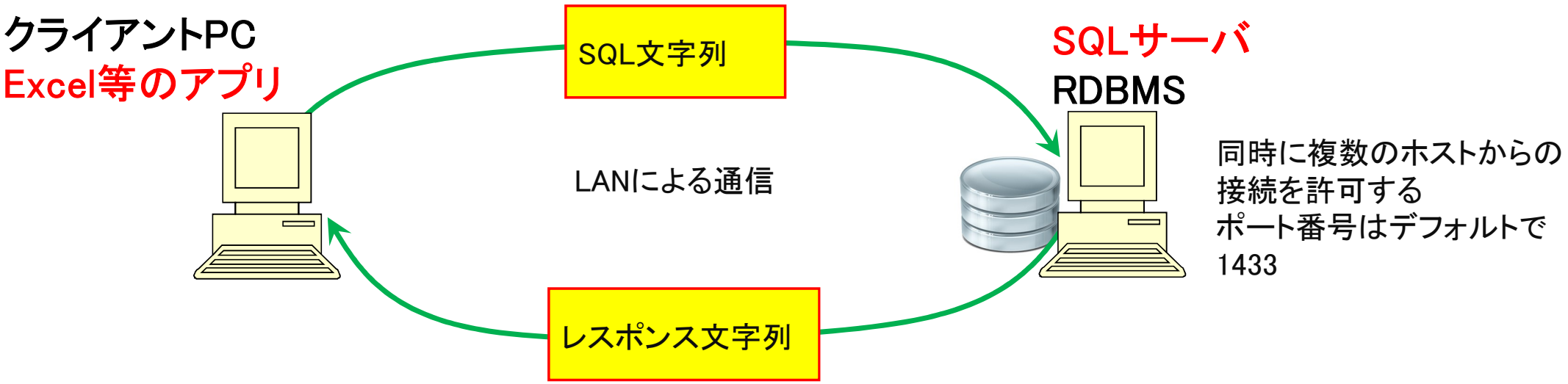
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

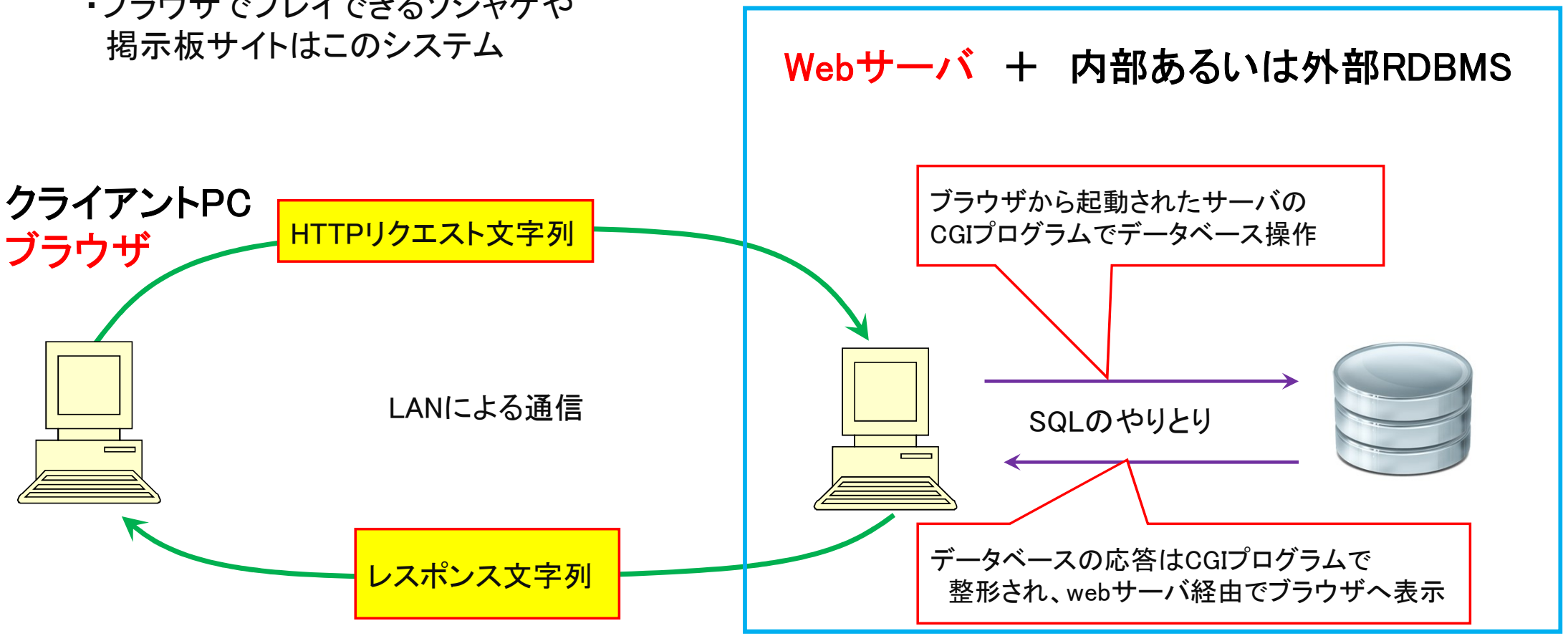
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

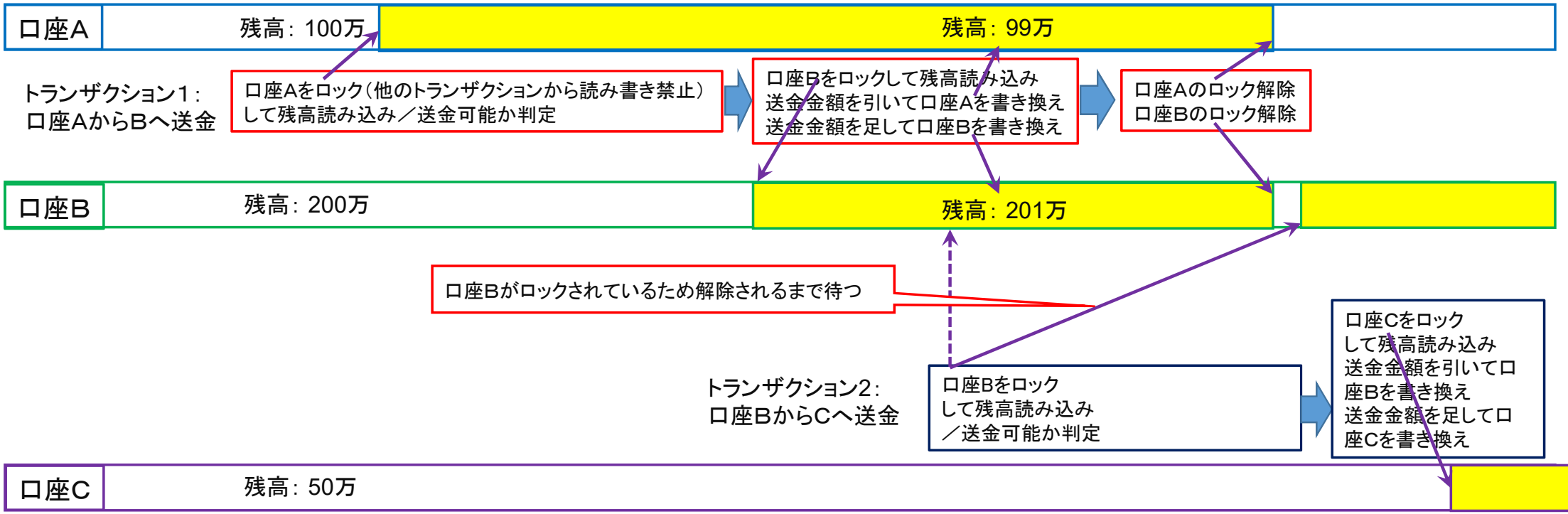
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

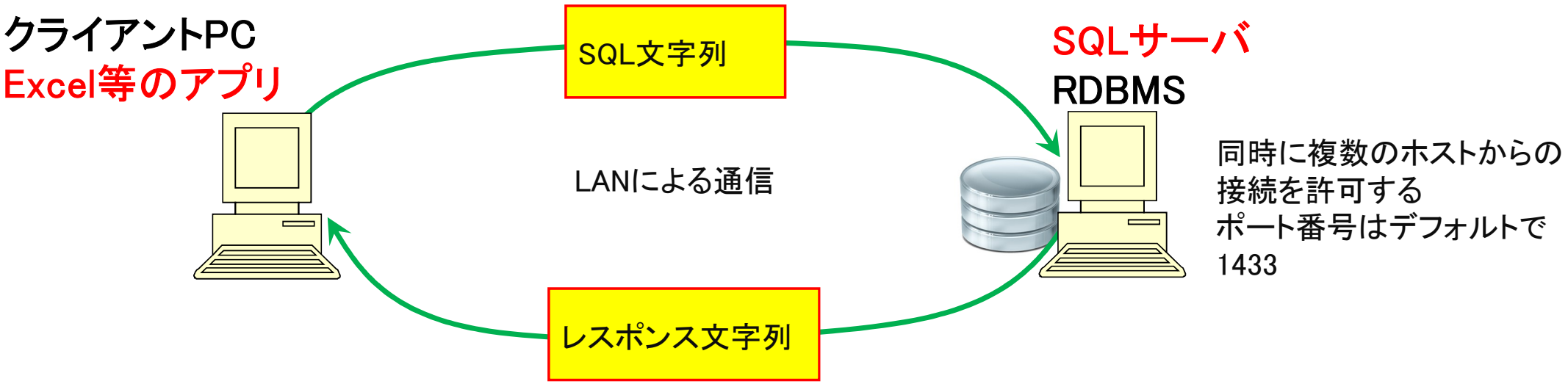
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

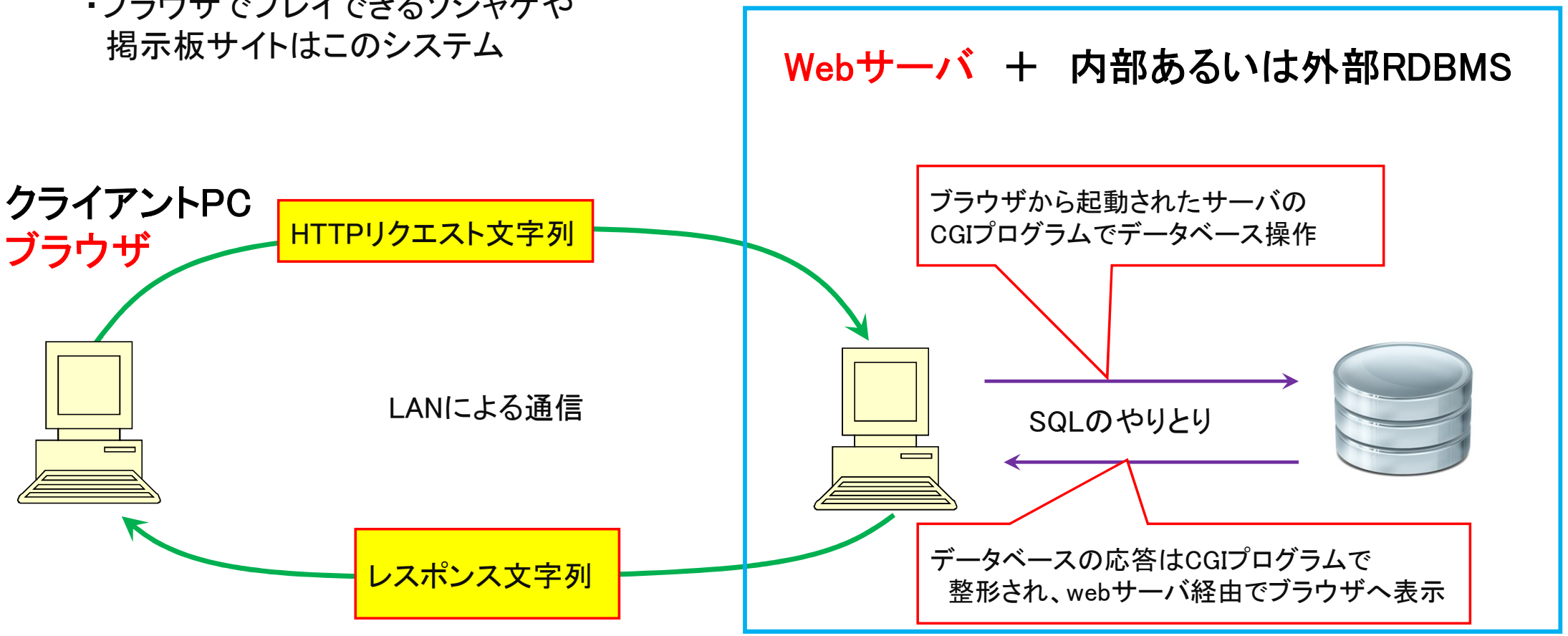
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

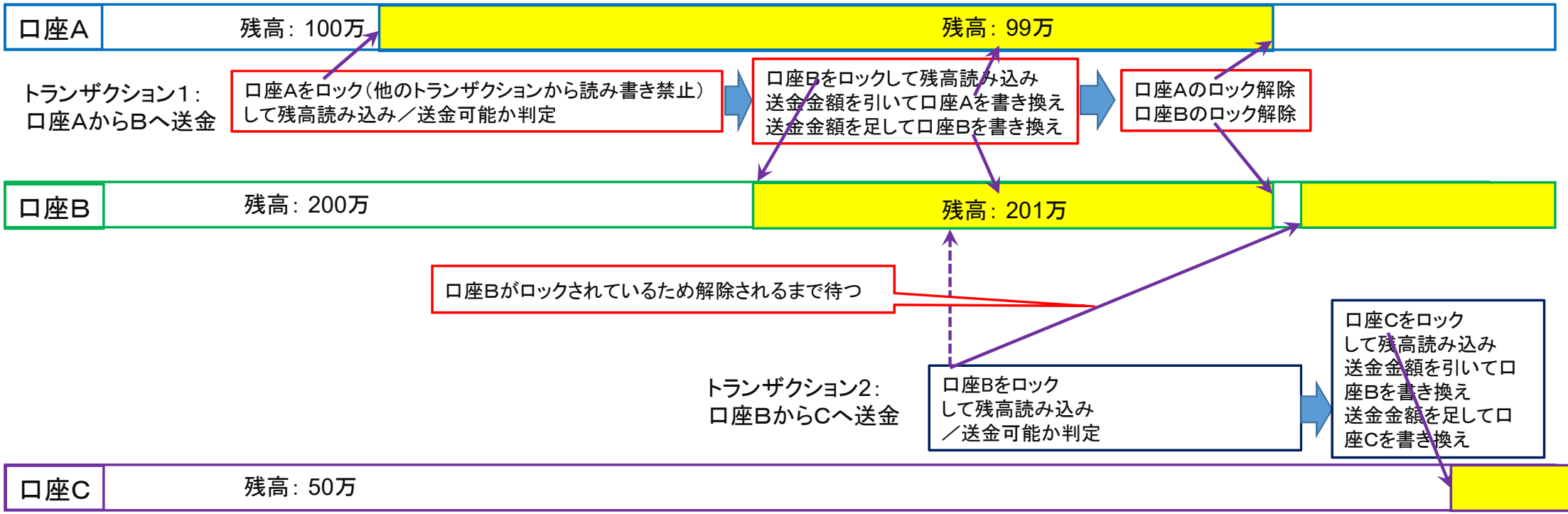
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

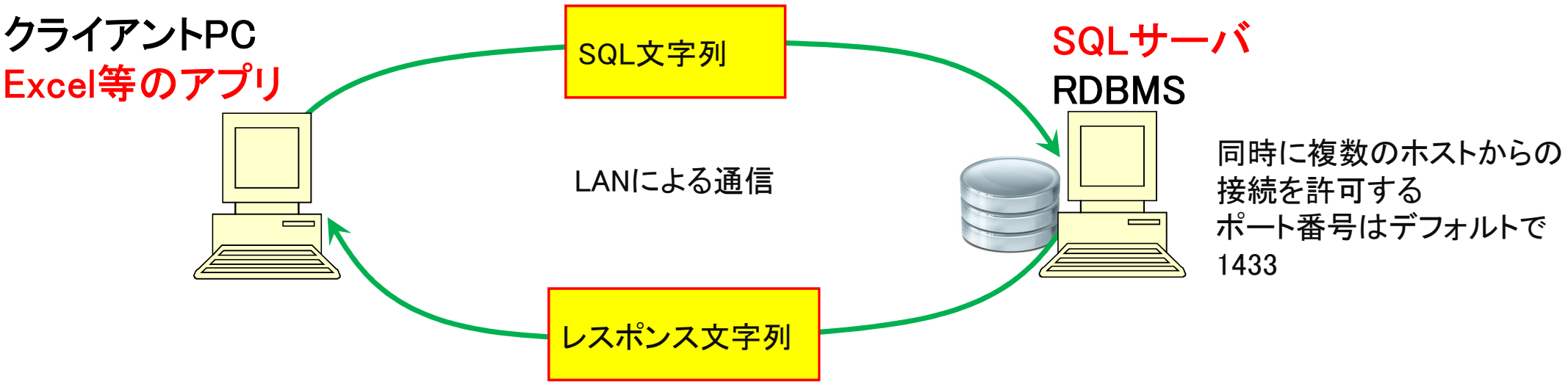
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

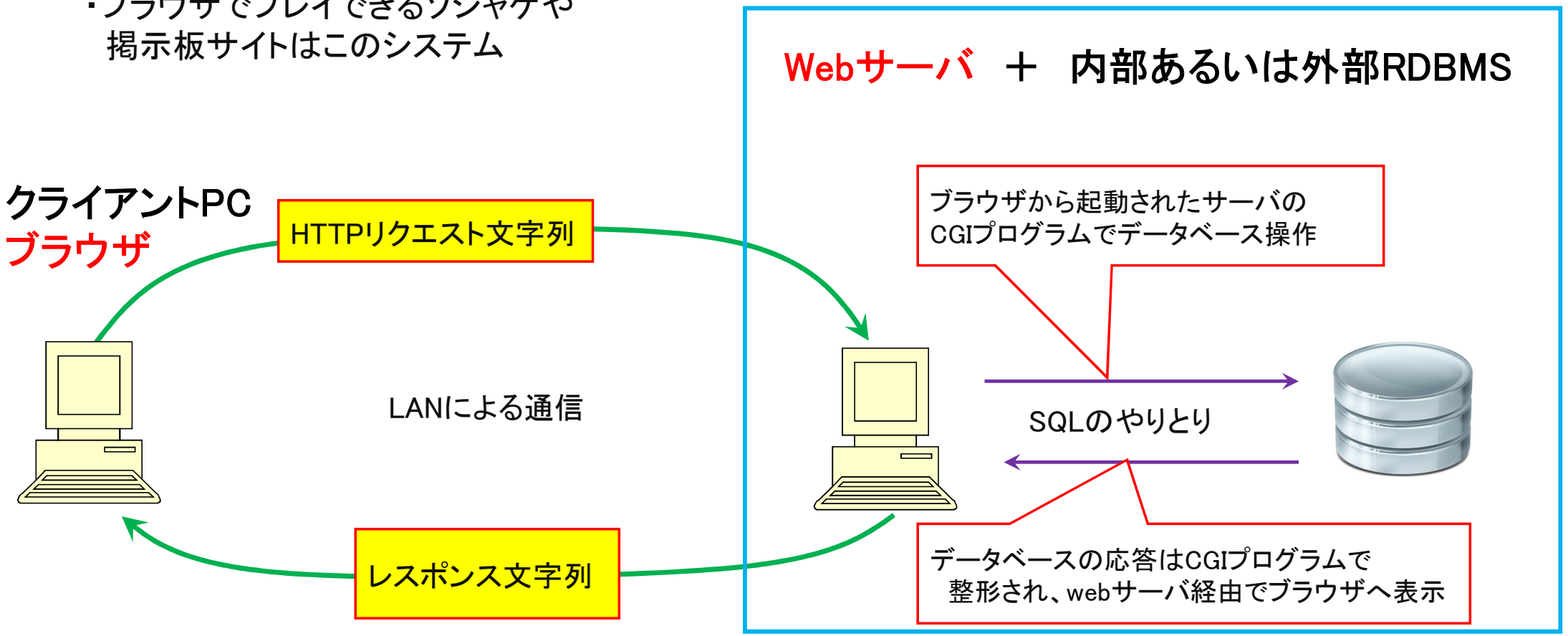
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

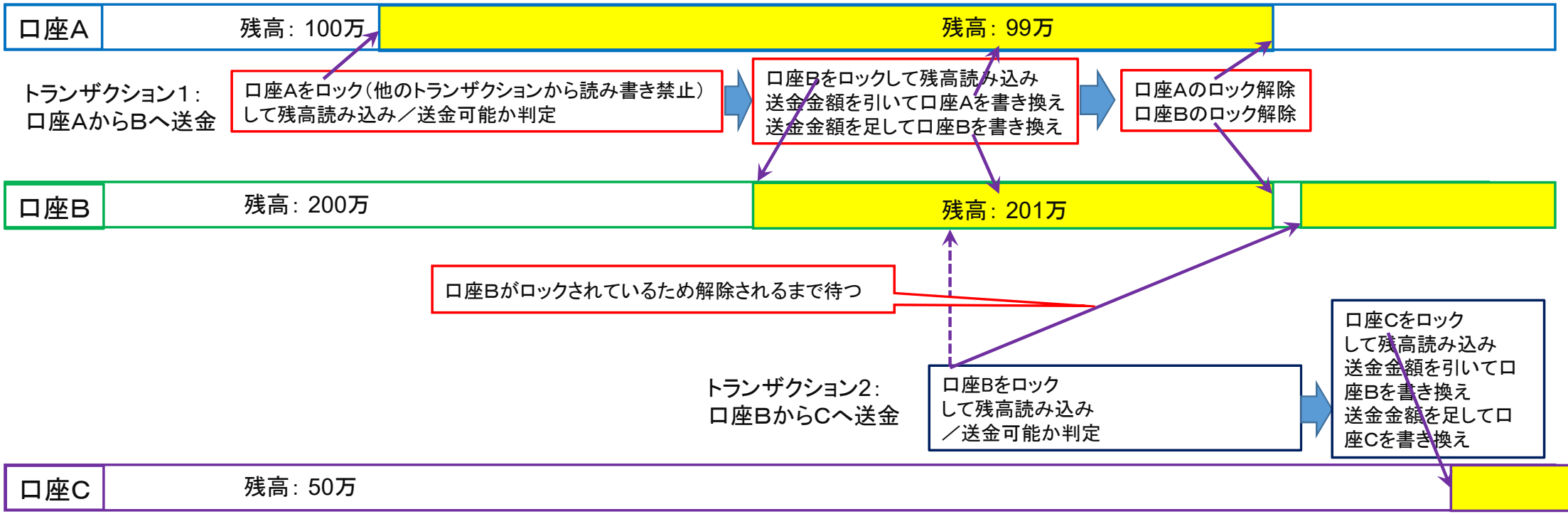
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

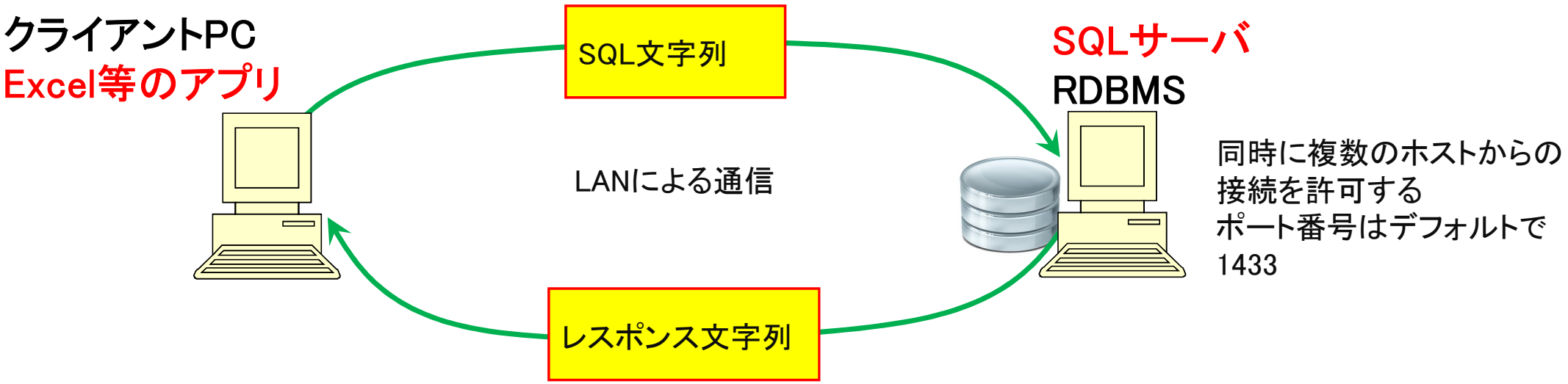
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

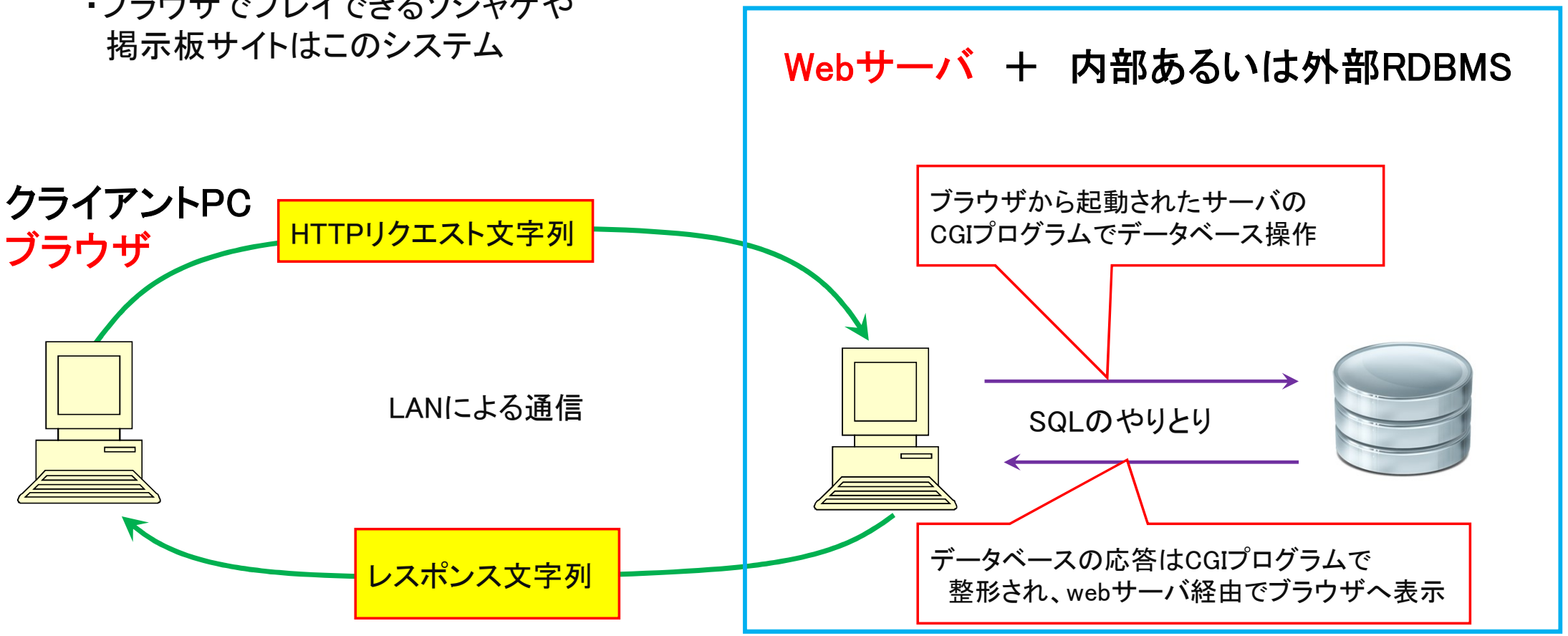
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

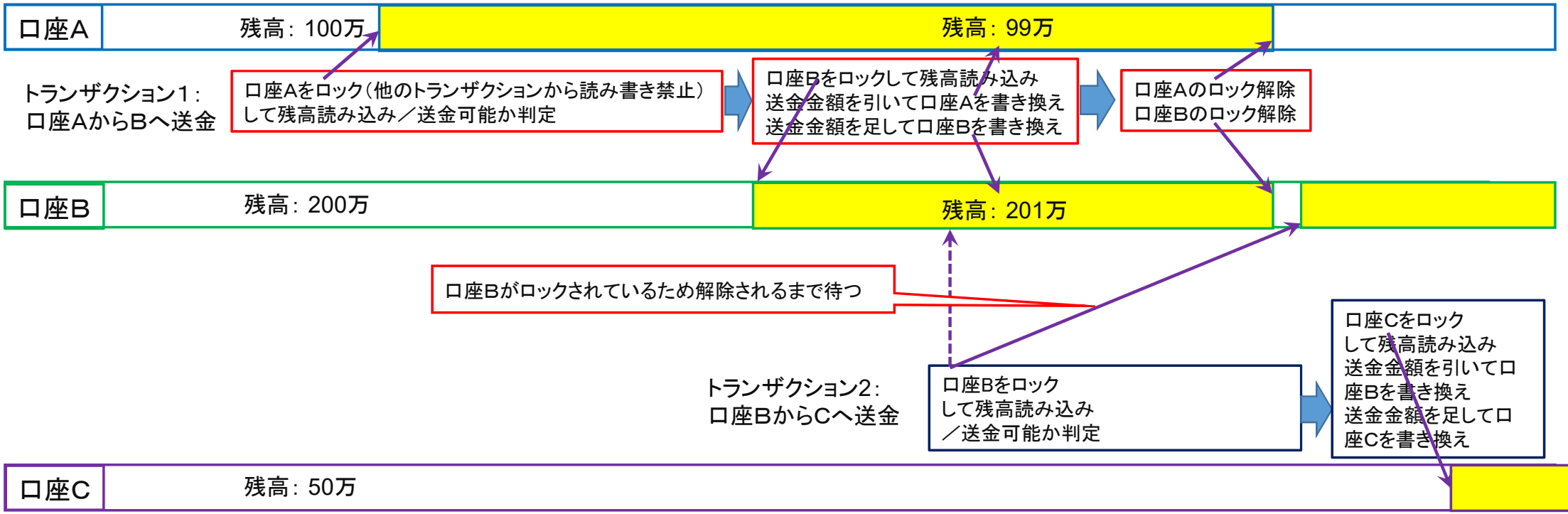
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

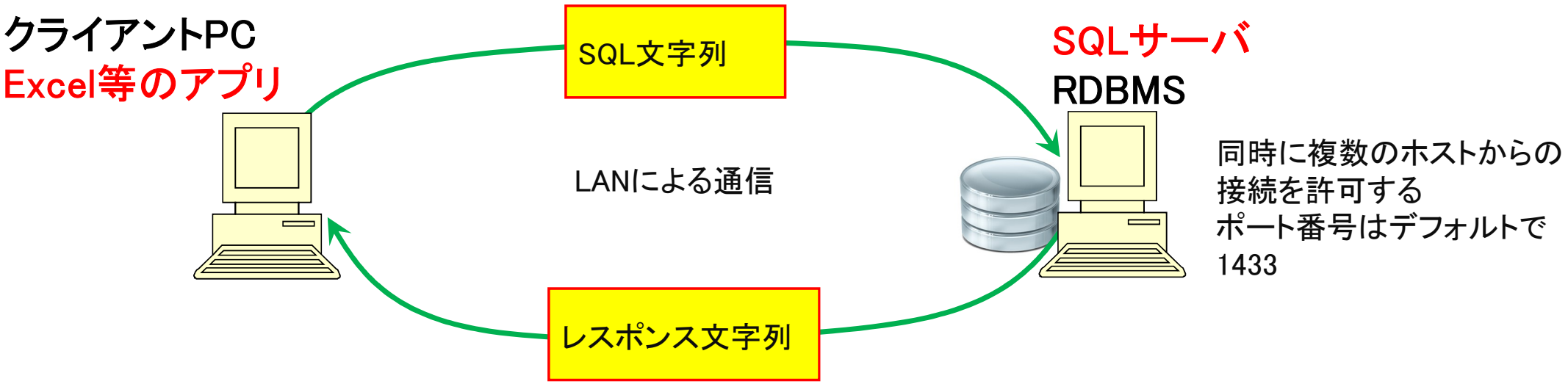
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

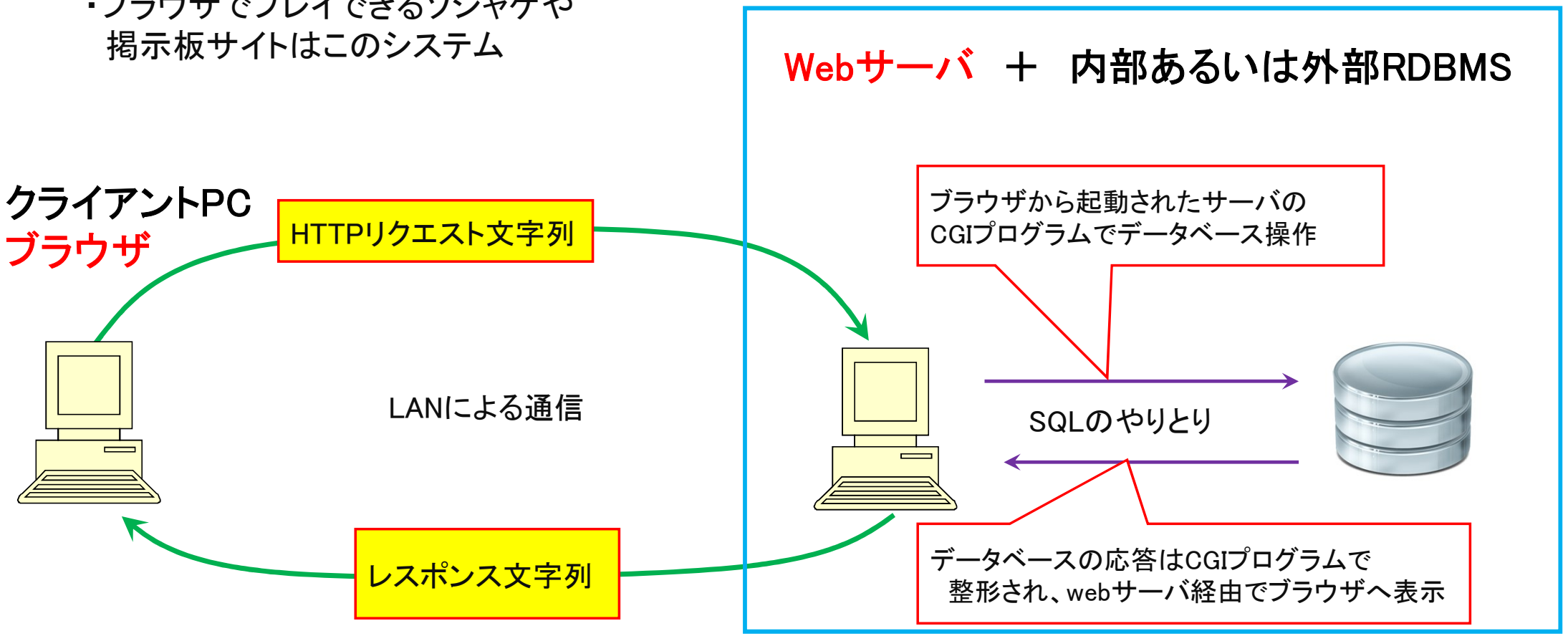
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

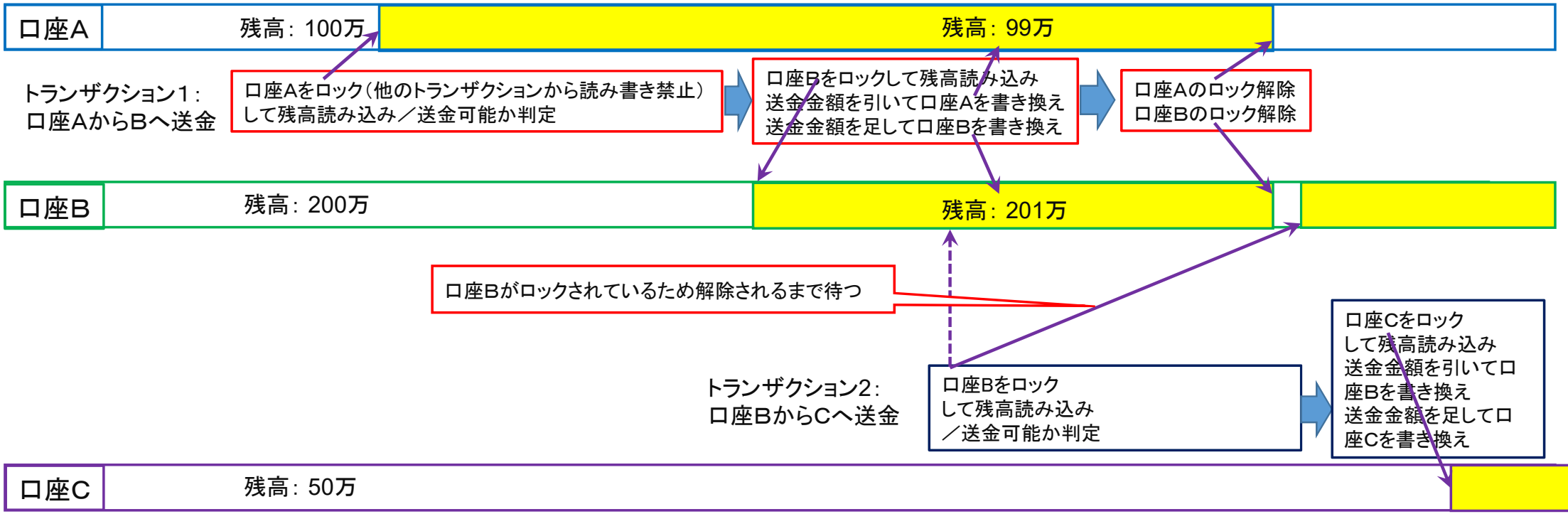
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

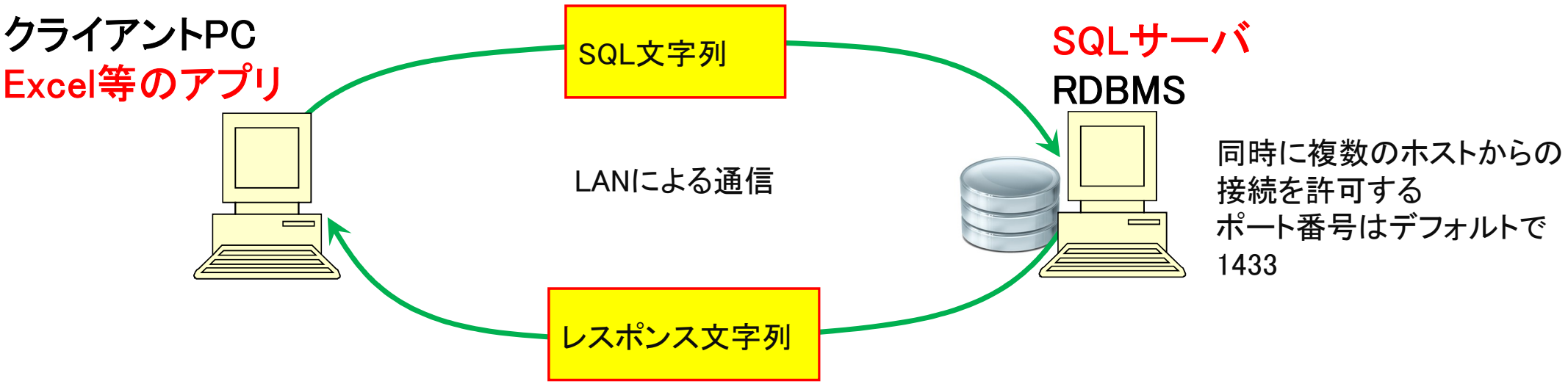
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

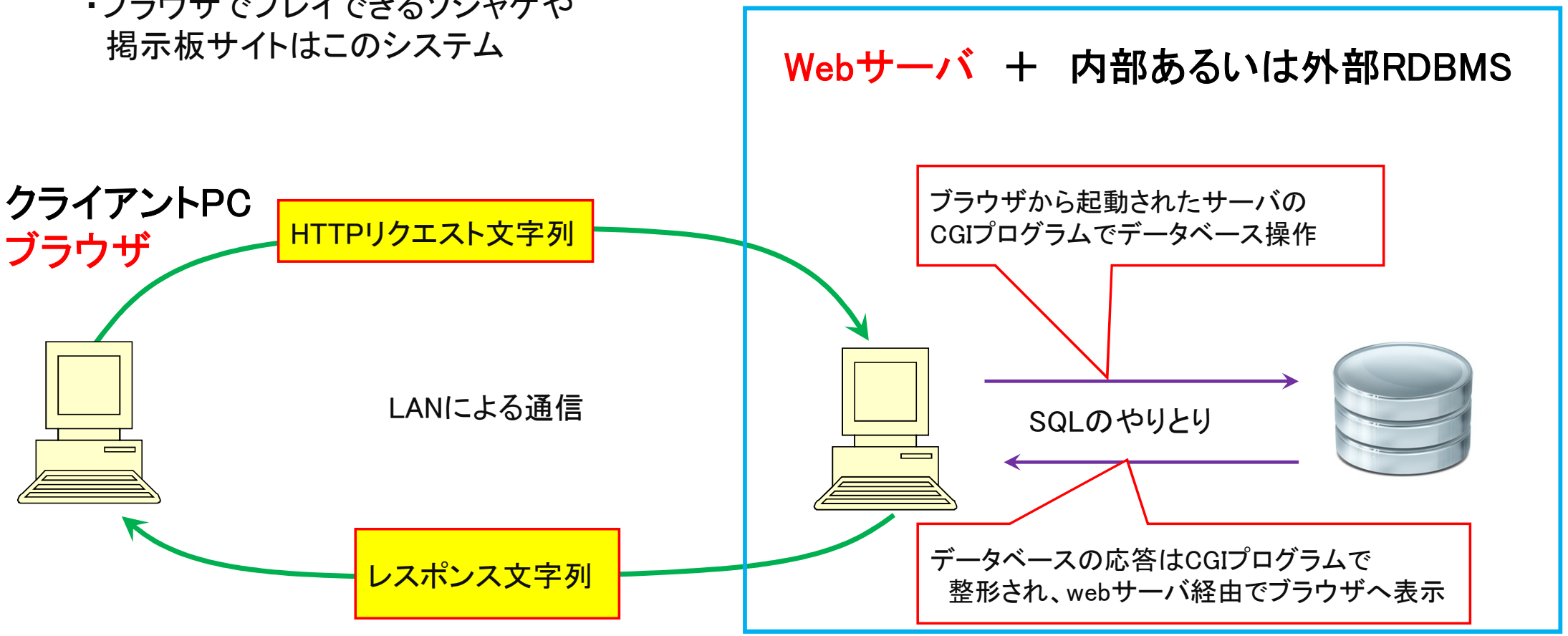
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

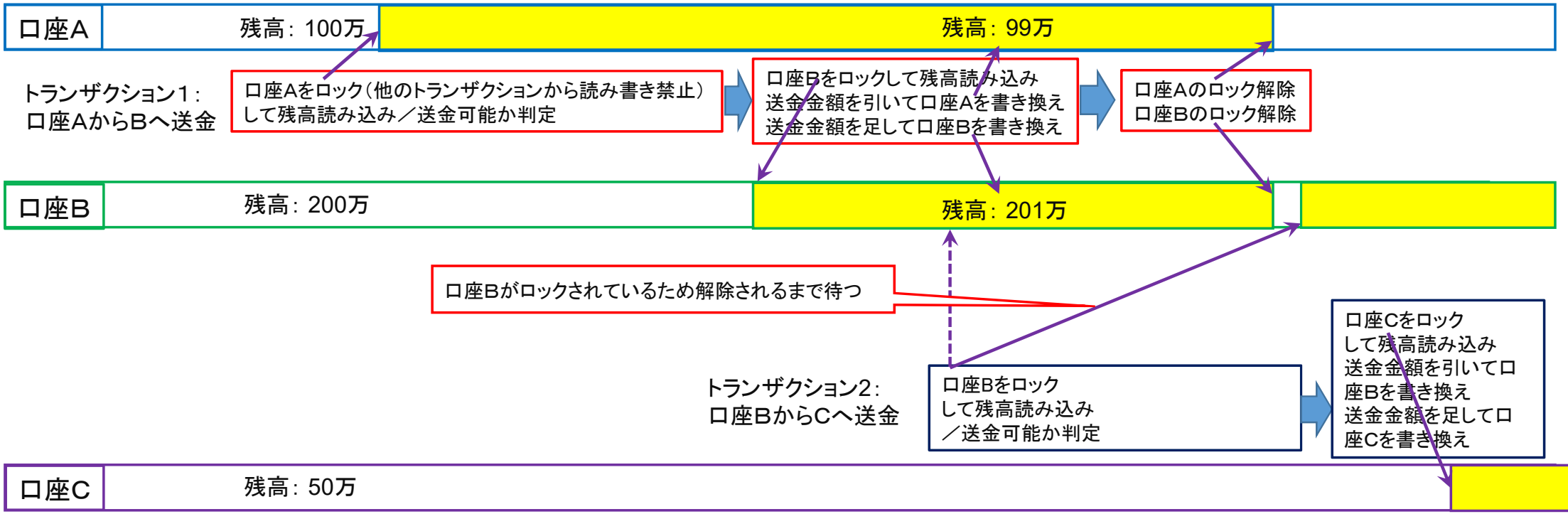
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

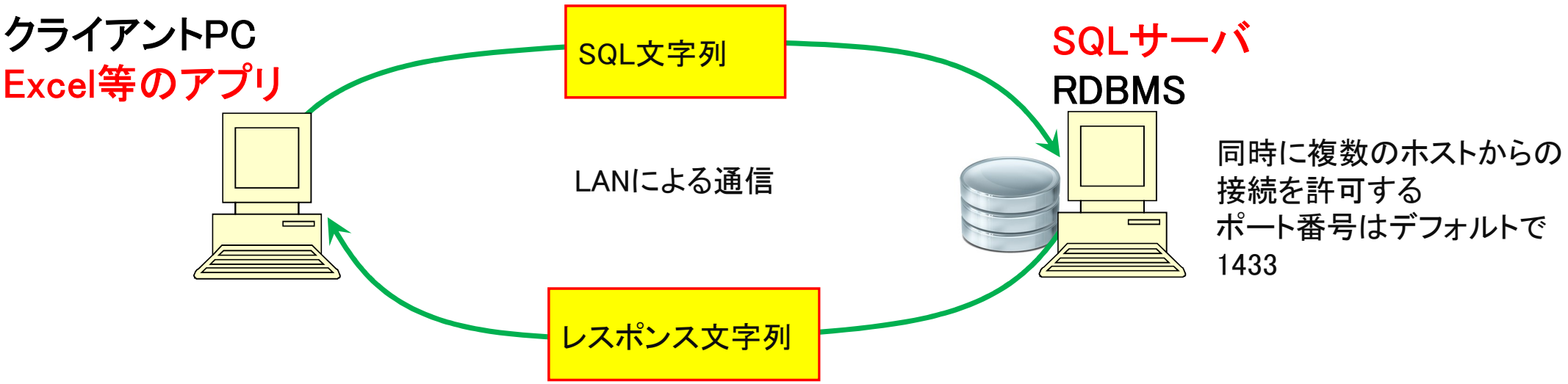
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

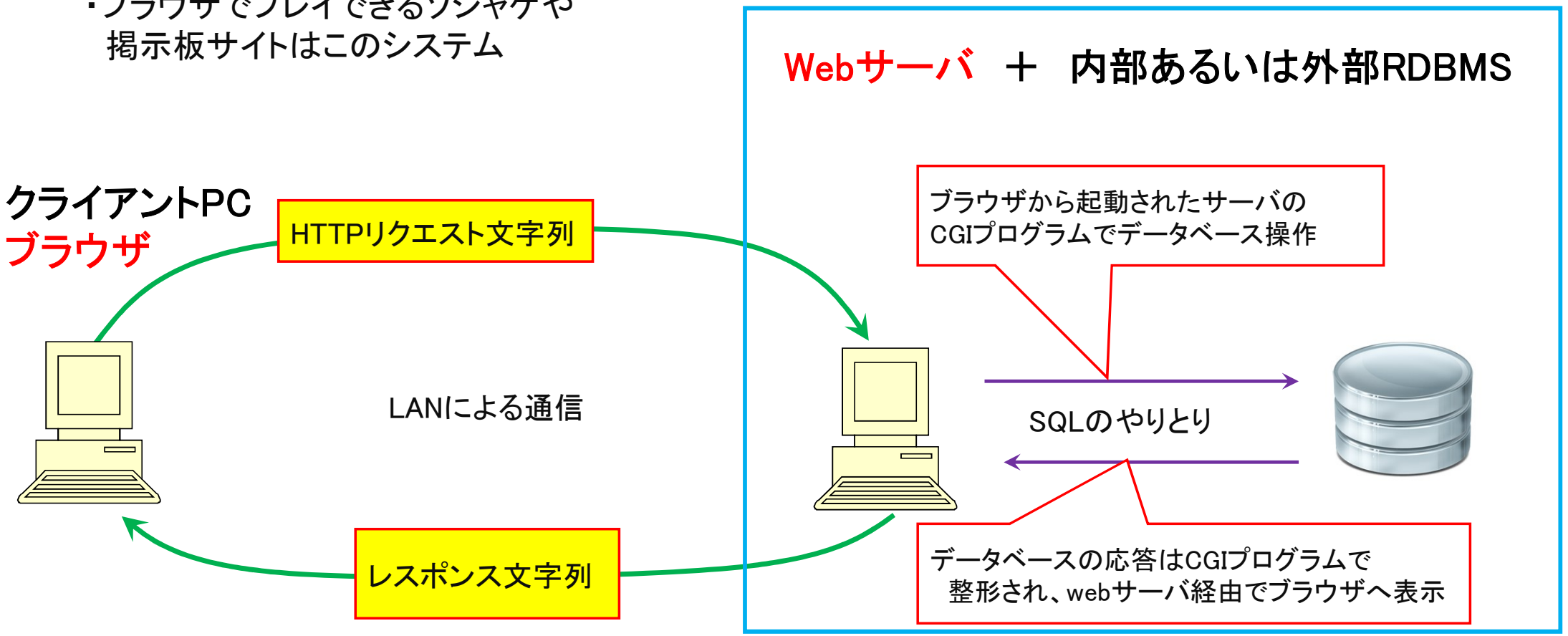
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い

船舶海洋情報学　07. データベース

九州大学　工学府海洋システム工学専攻　講義資料　担当：木村

九州大学
KYUSHU UNIVERSITY

「膨大な情報」は必要な情報を簡単に引き出せてこそ価値がある

- 何らかの規則を持ったデータの集まり
- それらのデータに対する「追加」や「検索」などの管理機能

データベース

実務ではまず間違いなく使う

【参考文献】基礎からのMySQL, 西沢夢路著，ソフトバンククリエイティブ株式会社

関係データモデル（Relational data model）

【参考文献】 ウィキペディア「関係データベース」
IBMのエドガー・F・コッドによって考案

1） 1つのデータベースには複数の「テーブル」が置かれる　「テーブル名」＝**関係変数**

例）顧客・販売管理データベース

属性（Attribute）
または列（カラム）

1件分のデータ＝「レコード」

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

1件分のデータ
＝「レコード」

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002	M4A3E8	M113	
…				

関係データベース（Relational database）

2） 目的に応じて複数テーブルにまたがるデータを**属性**で連結／求める表を得る

複数のテーブルで
共通の属性を使用

先の2つのテーブルを**「顧客番号」で関連付け**、顧客番号の代わりに「顧客氏名のデータ」を要求
→ データベース検索結果より得た**「顧客名別の売り上げ」**

顧客氏名	商品 1	商品 2	商品 3
Avril Haines	M1 Abrams		
John Smith	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	T34	T72	
Avril Haines	M4A3E8	M113	
…			

販売日を050116で限定して先の2つのテーブルを**「顧客番号」で関連付け**、
商品と送り先（顧客住所）のデータを要求 → データベース検索・演算結果より得た**「050116日の商品と送り先」**

顧客氏名	住所	商品 1	商品 2	商品 3
John Smith	Washington, D.C. 20505	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
Meroe Park	935 Pennsylvania Avenue, NW	T34	T72	

関係データベース（Relational database）

3） データの一貫性を保って維持管理

例） 顧客番号の変更処理

顧客番号0002を
9999へ変更

該当するレコード
修正

他のテーブルで同じ属性
を持つレコードも修正

「顧客データ」テーブル

顧客番号	顧客氏名	住所	電話番号	メール
0001	John Smith	Washington, D.C. 20505	1-703-482-0623	
0002→9999	Avril Haines	1000 Colonial Farm Rd, McLean	1-571-204-3800	
0003	Meroe Park	935 Pennsylvania Avenue, NW	1-202-324-3000	
…				

「販売データ」テーブル

販売日	顧客番号	商品１	商品２	商品３
051115	0002→9999	M1 Abrams		
051116	0001	FV4034 Challenger 2	M1A1 Abrams	Leopard 2
051116	0003	T34	T72	
051117	0002→9999	M4A3E8	M113	
…				

トランザクションとACID

トランザクション ： データベース操作において**分けることのできない一連の情報処理の単位**

例） 銀行の口座間送金

1） 口座Aの残高から1万円引く
2） 口座Bの残高に1万円を加える

この2つの操作は不可分
片方しか実行されなかったら、大問題
間に別の操作が入っても破綻する

データを守るため、信頼性のあるトランザクションシステムの持つべき性質：ACID

原子性 (ATOMICITY)	トランザクションに含まれるタスクが全て実行されるか、あるいは全く実行されないことを保証する性質
一貫性 (CONSISTENCY)	トランザクション開始と終了時にあらかじめ与えられた整合性を満たすことを保証する性質
独立性 (ISOLATION)	トランザクション中に行われる操作の過程が他の操作から隠蔽されること
永続性 (DURABILITY)	トランザクション操作の完了通知をユーザが受けた時点で、その操作は永続的となり、結果が失われないこと

例） 口座送金するかしないかのどちらかしかない

例） 口座Aの残高が負になる送金は操作できない

例）

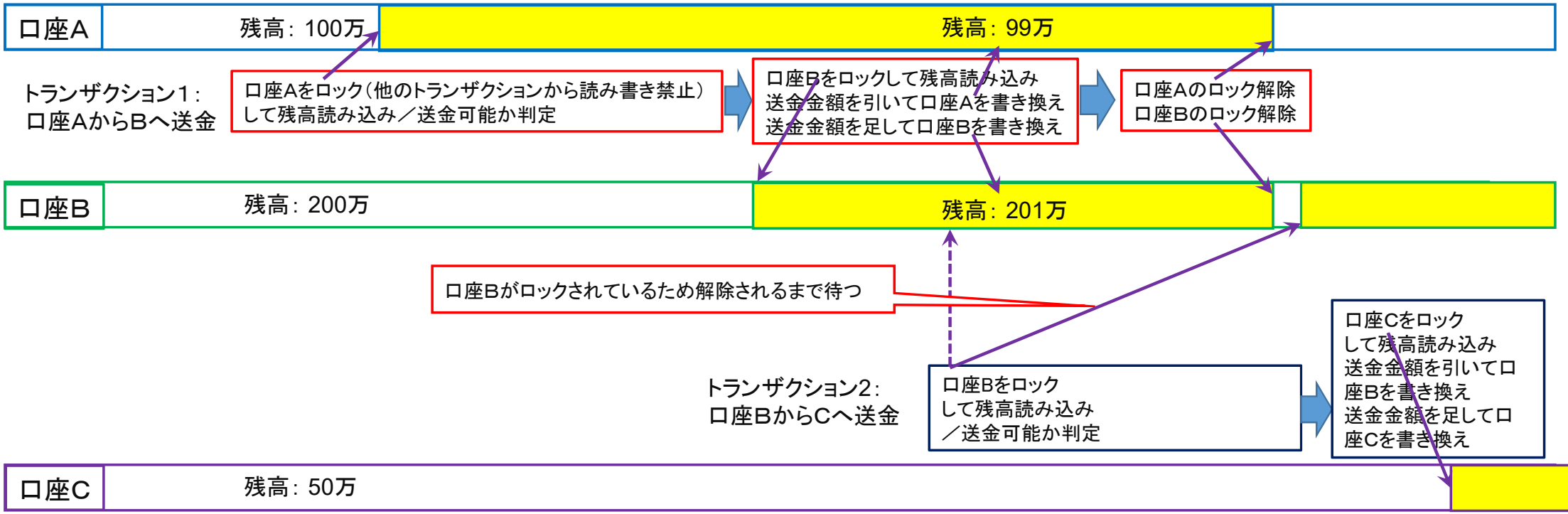
時点	口座 A	口座 B
送金前	100万	200万
実行中	99万	200万
送金後	99万	201万

外部から操作中の状態は見えない

トランザクションシステムがACIDを満たすための仕組み

複数のトランザクションとリソースのロック

データの一貫性を保つため、トランザクションは書き換えるリソースをロックする



必要なリソースが他のトランザクションにロックされている場合、解除されるまで待ち続けるとデッドロックすることがあるので
タイムアウトしてリソースを全て開始状態に戻してロックを解除してやりなおす

SQL（Structured Query Language）

・リレーショナルデータベース管理システム（RDBMS)に
問い合わせを行うためのコマンド言語。国際標準規格（SQL92,SQL99,JIS）

データベースのデータに対しての計算なども指示できる

・主なRDBMS

ソフト名称	特徴
Oracle	商用データベースシステムとして最も普及
Access	マイクロソフトOfficeファミリー（小規模）
Microsoft SQL Server	マイクロソフト社の商用ソフト
PostgreSQL	オープンソース
MySQL	オープンソース　世界で最も良く使われる

・データベース管理システムとのSQLのやりとりは、全て文字列（標準入出力）で行う

データベース単体ではとても使いにくい！

・データベース作成
CREATE DATABASE データベースの名前;

・データベースにテーブルを作成
CREATE TABLE テーブル名（カラム名１　データ型, カラム名２　データ型, …）;

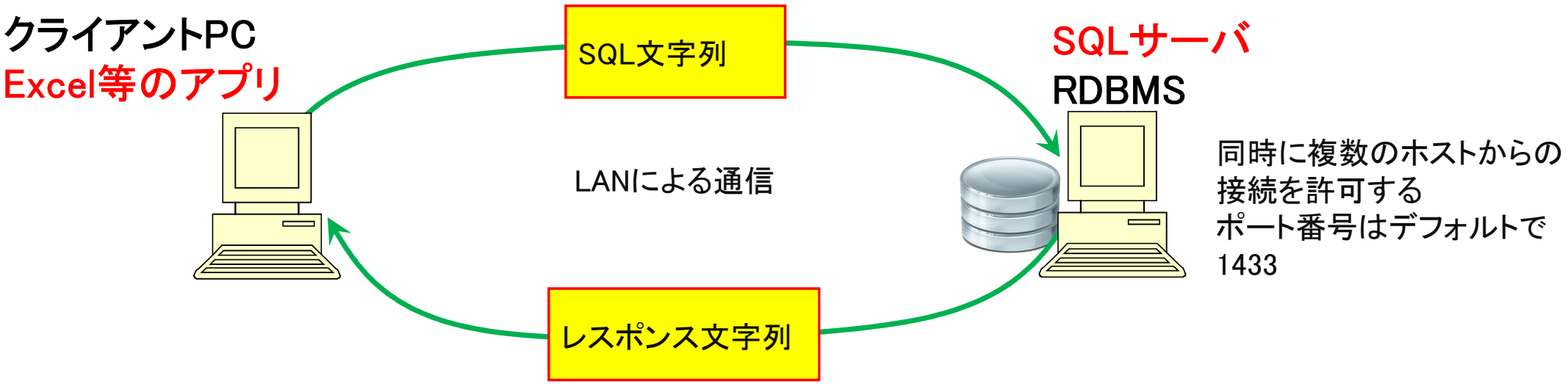
・テーブルにデータを入れる
INSERT INTO テーブル名　VALUES（データ1, データ2, …）;

・データを表示
SELECT カラム名１, カラム名２, … FROM テーブル名;

・データベースからは、実行結果がテキストで返ってくる

データベースがよく使われる形態1： TCP接続によるSQLサーバシステム

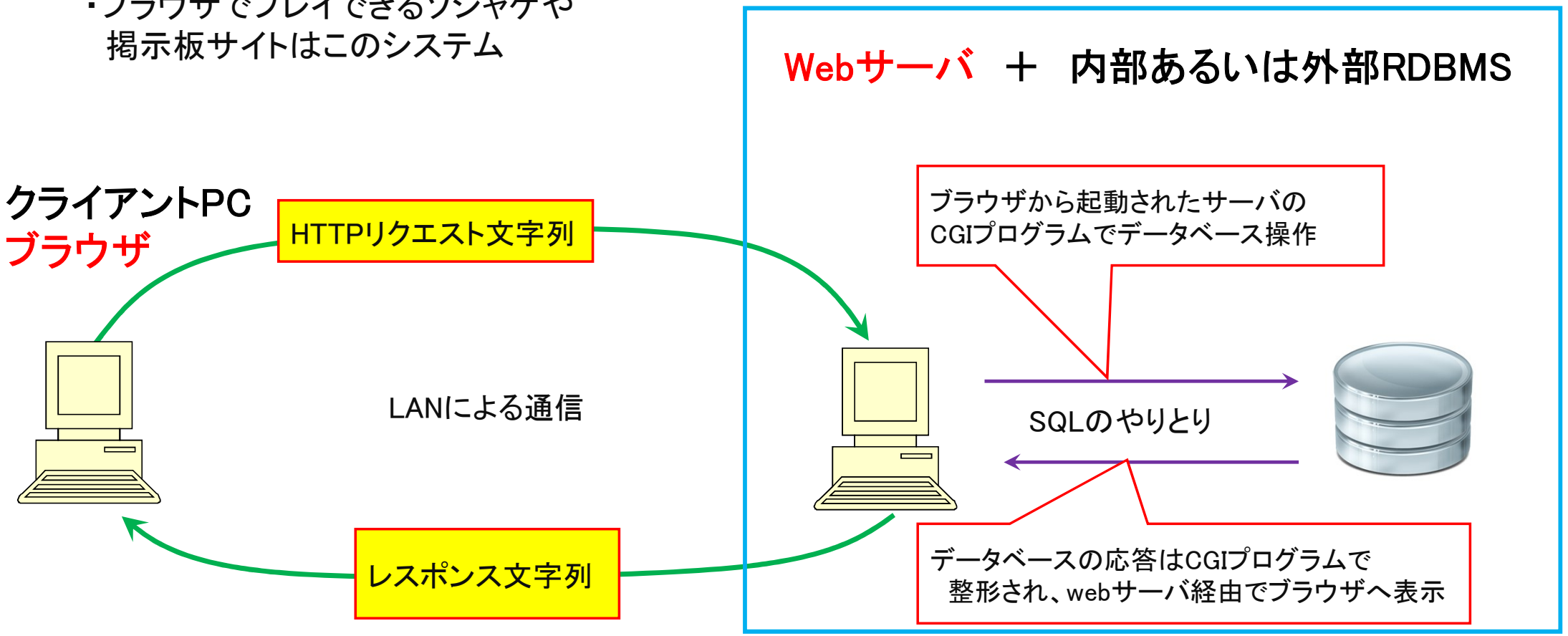
・リレーショナルデータベース管理システム（RDBMS）は，直接人間を相手にすることは無く、
プログラム等と通信する場合がほとんど



データベースがよく使われる形態2：　Webサーバと結合・ブラウザで操作閲覧

・CGIに用いられるプログラム言語にはSQLサーバとやりとりするための手段が用意されている
　PHP(PDO)，Python (pyodbc), Java,

・ブラウザでプレイできるソシャゲや
　掲示板サイトはこのシステム



カラム（列）のデータ型

・データベースのテーブルは、それぞれのカラムに指定した形式のデータしか入力できない

数値型

INT:　整数
DOUBLE:　倍精度浮動小数点
など

これら以外にも、
通貨型などもある

文字列型

CHAR:　255文字までの文字列
TEXT:　65535文字までの文字列
など

日付・時刻型

DATETIME:　日付と時刻 9999-12-31-23:59:59

バイナリ型

binary[n]:　nバイトの固定長ナイナリデータ
varbinary[(n | max)]　最大格納サイズnバイトの可変長バイナリデータ

データファイルをそのまま
データベースのカラムへ突っ込める

SQLデータベースの実例

・JASNAOE講演会の論文投稿システム

Webのフォームから著者氏名・所属・講演タイトル・分野（複数）・アブストラクト・メールアドレスを入力

データベースへ登録

分野別に検索され講演会プログラム作成に利用

・データベース運用における実務上の注意点

データベースのテーブル作成時、複数のテーブル間で、**同じ意味を持つ属性があっても、誤って異なる名称を付けてしまった**ために関連付けできなくなってしまう事例が多数発生

例）「線図」「ラインズ」「LINES」「Lines」など全て異なる属性名称

一応**SQLシステム上は置き換えやマージ操作は可能**だが、実務上事例が多すぎて直す人手が足りない

まとめ

（1） 関係データベースモデル

（2） 1件分のデータ＝レコード
「テーブル」　＝　エクセルのシート、
「複数のテーブル」 ＝　エクセルのブック、
「カラム（列）」:属性　＝　エクセルのセルの縦一列　のイメージ

（3） **トランザクション**: 分けることのできない一連の情報処理の単位
データベースは複数同時にアクセスがあってもデータの一貫性を保つ堅牢なシステム

（4） **SQL**:　関係データベースシステムを操作するコマンド言語で国際標準規格

（5） データベースはSQLサーバとして運用し、他のアプリやプログラムを通じてデータを出し入れ
データベースを単体で直接操作することはほとんど無い