
PERFORMANSE
MIRRORING
AVAILABILITY
TYPE
DESIGNER
NETWORK
RELATIONSHII
LAYOUT
OPTIMIZATION
SYSTEM
QUERY
OBJECT
MIGRATION
SERVER
RELATIONAL
DBMS
ARCHITECTURE
ENTITY
DATA
DATABASE
STORAGE
CONTENT
BUSINESS
DEVELOPER
DESIGN
TABLE
SECURITY
RECOVERY
SOFTWARE
SQL
XML
BACKUP
KEY
RESTORE
DEVELOPMENT
ADMINISTRATION
INDEX
LANGUAGE
COLLECTION
STRUCTURE
TRANSACTION
CONCEPT
REPLICATION

Somkiat Puisungnoen

Somkiat | Home 17

Somkiat Puisungnoen

Update Info | View Activity Log 10+ | ...

Timeline | About | Friends 2,941 | Photos | More

When did you attend Ubon Ratchathani University?

14 Pending Items

Intro

Software Craftsmanship

- Software Practitioner at สยามชำนาญกิจ พ.ศ. 2556
- Agile Practitioner and Technical at SPRINT3r
- Software analyst at TARAD.com
- Software Developer at True Corporation
- Former Software Engineer at Opendream

Status | Photo/Video | Live Video | Life Event

What's on your mind?

Public | Post

Somkiat Puisungnoen
1 hr · Twitter ·

พรุ่งนี้ไป share เรื่อง NoSQL ที่มหาวิทยาลัย

Like | Comment | Share

Nuttachot Dusitanont, Chitpong Few Wuttanan and 50 others

somkiat.cc

Somkiat | Home

Page | Messages | Notifications | Insights | Publishing Tools | Settings | Help

somkiat.cc

@somkiat.cc

- Home
- About
- Events
- Photos
- Likes
- Videos
- Posts
- Services

Liked | Following | Share | ...

+ Add a Button

Write something...

Personal Blog

Page Tips — See All

What's a Boosted Post?
A boosted post is the easiest way to reach more people on Facebook.

SQL (Structured Query Language)

Pronounced:

se·quel ('sēkwəl/)

A **declarative** language for asking questions from a **relational** database, invented in *1974!*

SQL Standard

SQL 92

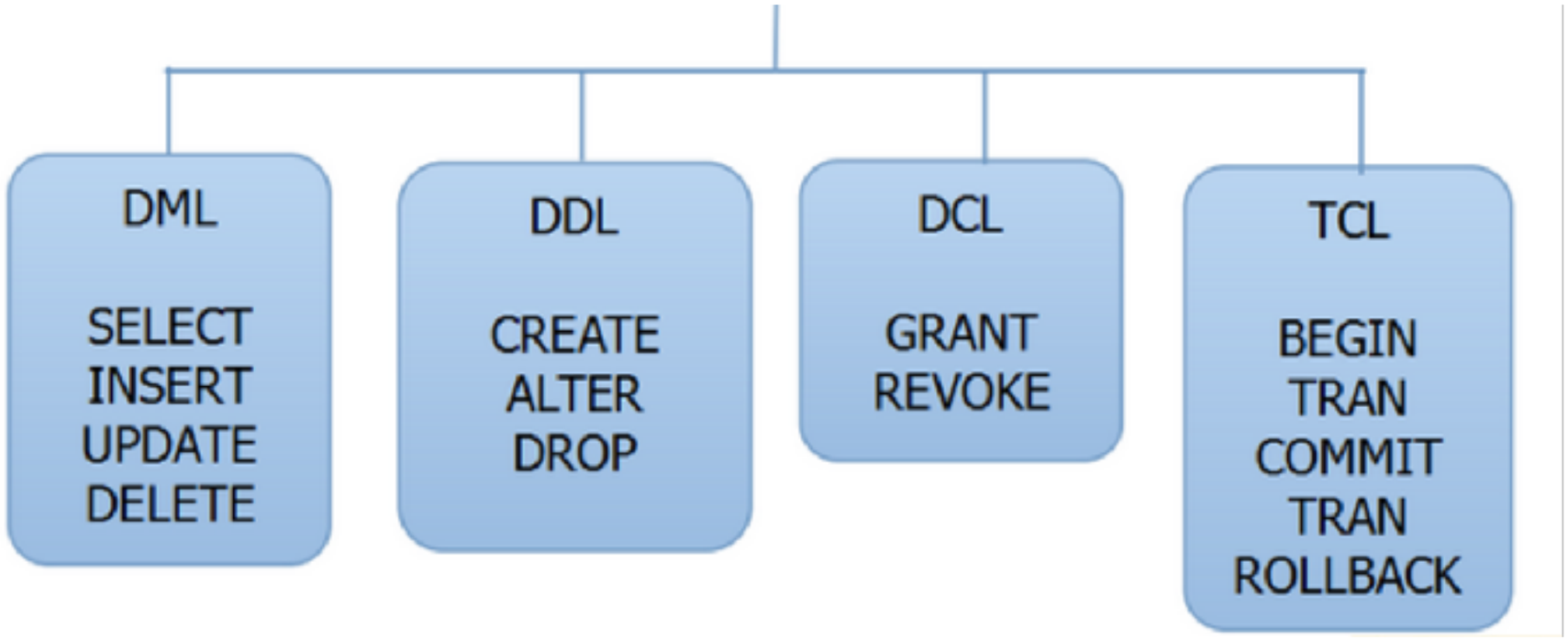
SQL 99

SQL 2013

SQL Statements

Data Definition Language (DDL)
Data Manipulation Language (DML)
Data Control Language (DCL)
Transaction Control Language (TCL)
Data Retrieval Language (DRL)

SQL Statements



Data Definition Language (DDL)

Define the db structure or schema

- Create
- Alter
- Drop
- Truncate
- Rename

Data Manipulation Language (DML)

Manage data with in schema objects

Insert
Delete
Update
Merge

Data Control Language (DCL)

Control the level of access that user have

Grant
Revoke
Deny
Constraints

Transaction Control Language (TCL)

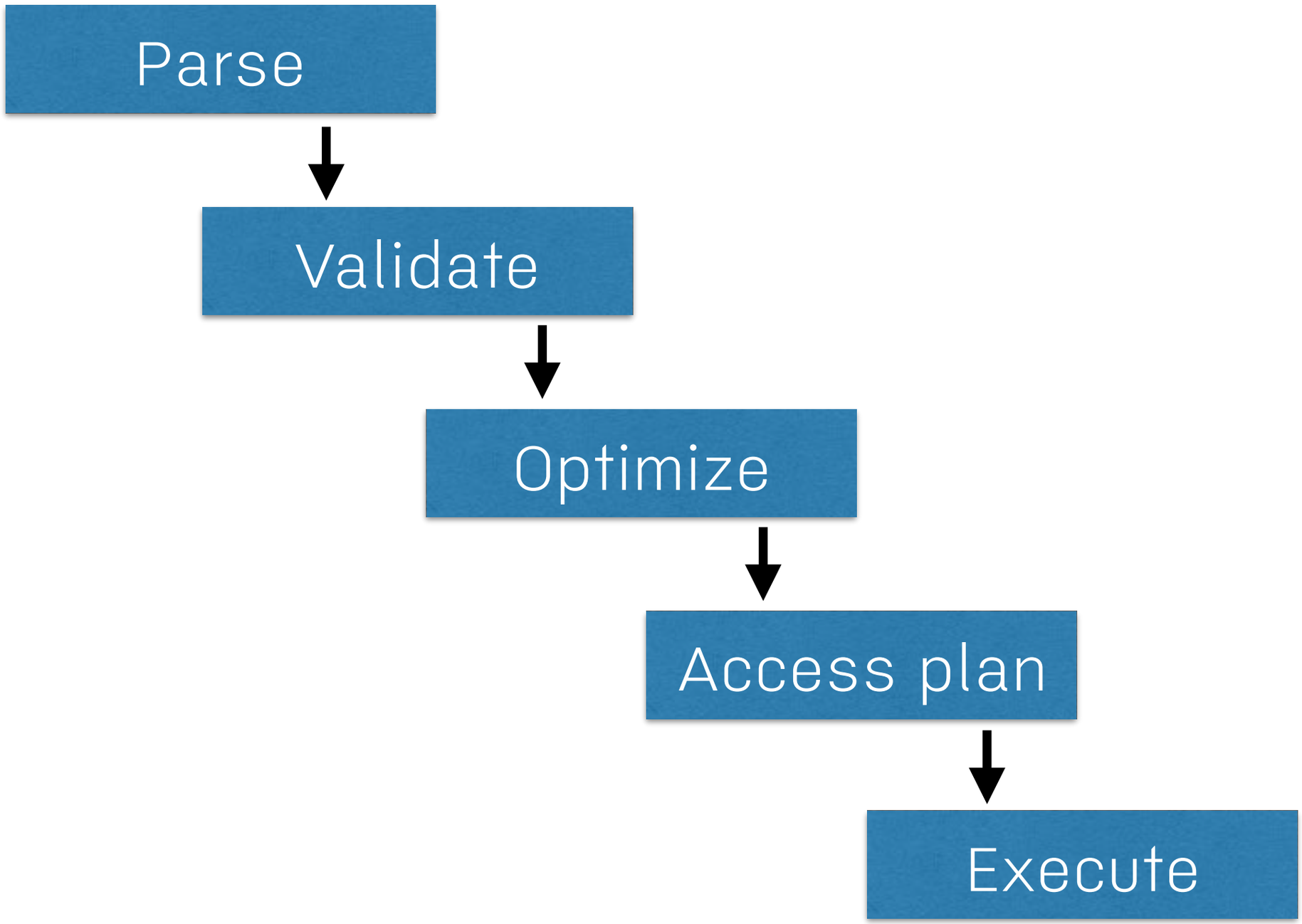
Control and manage transaction to maintain the integrity of data

Begin
Commit
Rollback
Savepoint

Data Retrieval Language (DRL)

Select

SQL Statement Processing



Data Definition Language (DDL)

Manage Database Structure

Create
Alter
Drop
Truncate
Rename

Working with MySQL

Working with MySQL

$mysql -u <user> -p <password>

Working with MySQL

$show databases;
$use <database name>;
$show tables;
$desc <table name>;
$show create table <table name>

1. Create Table

Use **create** statement

Specify **columns** with data type and columns constraints

Specify **table constraints**

- Primary key (PK)
- Foreign key (FK)

Basic Syntax

```
CREATE TABLE <table name>
(
    <column name> <data type> <constraints>
   ,<column name> <data type> <constraints>
    ...
    <table constraints>
);
```

SQL Data Type

Data type	Access	SQLServer	Oracle	MySQL	PostgreSQL
boolean	Yes/No	Bit	Byte	N/A	Boolean
integer	Number (integer)	Int	Number	Int Integer	Int Integer
float	Number (single)	Float Real	Number	Float	Numeric
currency	Currency	Money	N/A	N/A	Money
string (fixed)	N/A	Char	Char	Char	Char
string (variable)	Text (<256) Memo (65k+)	Varchar	Varchar Varchar2	Varchar	Varchar
binary object	OLE Object Memo	Binary (fixed up to 8K) Varbinary (<8K) Image (<2GB)	Long Raw	Blob Text	Binary Varbinary

Summary table

Column name	Column type	Column Constaints
isbn	varchar(20)	primary key

Create summary table

```
create table summary(
   isbn varchar(20) primary key
);
```

2. Modify Table

Add columns
Delete columns
Rename columns
Add column constraints
Add table constraints

Basic Syntax

```
ALTER TABLE <table name>
ADD <column name>,
ADD <table constraint>
MODIFY <column name>
```

Summary table

Column name	Column type	Column Constaints
isbn	varchar(20)	primary key
amount	decimal(5,2)	

Create column amount

```
ALTER TABLE summary
ADD amount decimal(5,2);
```

Summary table

Column name	Column type	Column Constaints
isbn	varchar(20)	primary key
amount	int	

Change data type of column

```
ALTER TABLE summary
MODIFY amount int;
```

Summary table

Column name	Column type	Column Constaints
isbn	varchar(20)	primary key
amount2	int	

Change name of column

```sql
ALTER TABLE summary
CHANGE amount amount2 int;
```

Column Constraints

Primary key
Not NULL
CHECK clause
Default
Unique

Summary table

Column name	Column type	Column Constaints
isbn	varchar(20)	primary key
amount2	int	>= 0

Add column constraints

```
ALTER TABLE summary
ADD constraint check (amount2 >= 0);
```

Table Constraints

Primary key
Foreign Key
Index

Summary table

Column	Column type	Column Constaints	Table Constraints
isbn	varchar(20)	primary key	FK to isbn of book
amount2	int	>= 0	

Book table

Column	Column type	Column Constaints	Table Constraints
isbn	varchar(20)	primary key	

Add column constraints

```
CREATE TABLE book(
   isbn varchar(20) primary key
);
```

```
ALTER TABLE summary
ADD CONSTRAINT fk_isbn
FOREIGN KEY (isbn) REFERENCES book(isbn);
```

Add column constraints

```sql
// 1. Add Primary KEY
ALTER TABLE summary
ADD PRIMARY KEY(isbn);

// 2. Add FOREIGN KEY
ALTER TABLE summary
ADD CONSTRAINT fk_isbn
FOREIGN KEY (isbn) REFERENCES book(isbn);

// 3. Add INDEX
ALTER TABLE `table`
ADD INDEX `index nane` (`product_id`)
```

Delete column constraints

```sql
// 1. Delete Primary KEY
ALTER TABLE summary
DROP PRIMARY KEY;

// 2. Delete Foreign KEY
ALTER TABLE summary
DROP FOREIGN KEY fk_isbn;

// 3. Delete INDEX
ALTER TABLE `table_name`
DROP INDEX  `id_name_fk`;
```

3. Create View

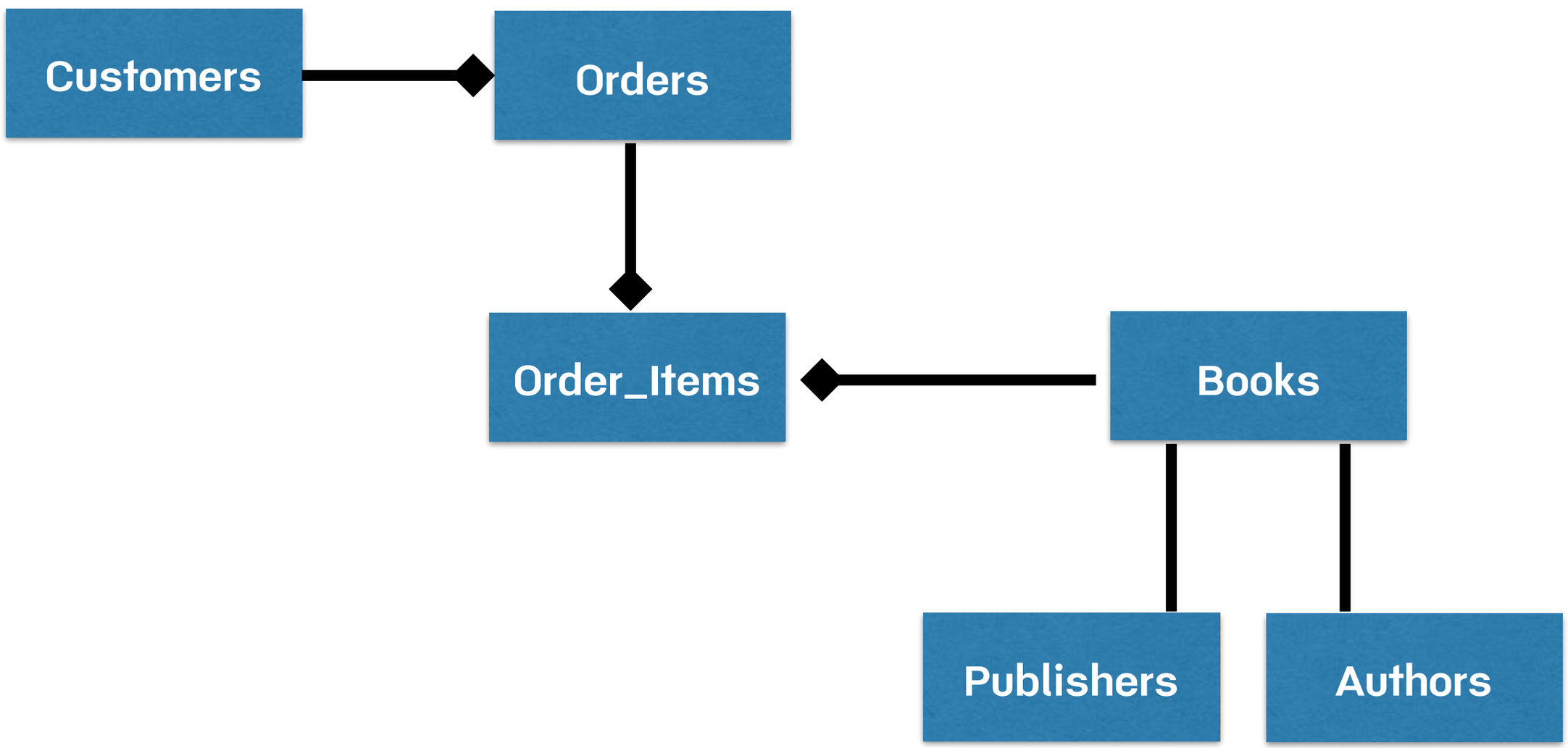
Named query is stored in the database
Can be read like a table

Basic Syntax

```
CREATE VIEW  <view name>
( <column name>, <column name>, ...  )
AS
<Select stetment>
```

Workshop

E-commerce :: Book Store



Customers

Column name	Column type	Column Constaints	Table Constraints
id	int	primary key	
first_name	varchar(20)	NULL	
last_name	varchar(50)	NULL	
address	varchar(100)	NULL	
phone_no	varchar(20)	NULL	

Books

Column name	Column type	Column	Table Constraints
id	int	primary key	
isbn	varchar(20)	primary key	
title	varchar(200)	NULL	
price	decimal(10, 2)	NULL	
publisher_year	int	NULL	
publisher_id	int	NOT NULL	FK refer to Publishers
author_id	int	NOT NULL	FK refer to Authors

Orders

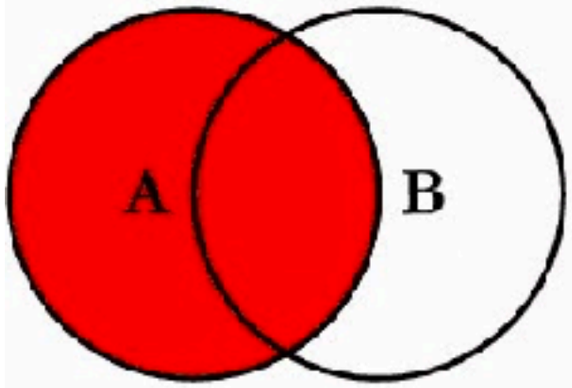
Column name	Column type	Column Constaints	Table Constraints
id	int	primary key	
customer_id	int	NOT NULL	FK refer to customer
total_price	decimal(10, 2)	NULL	
order_status	int	NULL	
create_datetime	datetime	NOT NULL	

Order_items

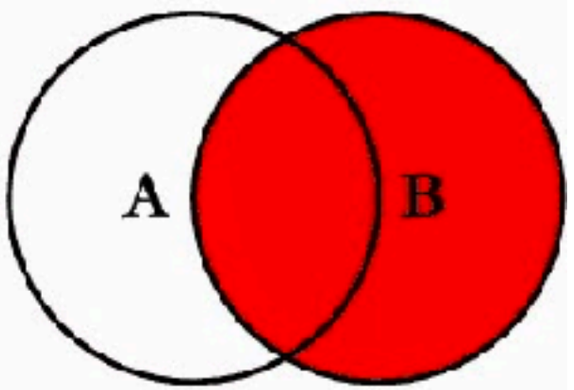
Column name	Column type	Column Constaints	Table Constraints
id	int	primary key	
order_id	int	NOT NULL	FK refer to orders
quantity	int	NULL	
unit_price	decinal(10,2)	NULL	
create_datetime	datetime	NOT NULL	

Workshop with CRUD

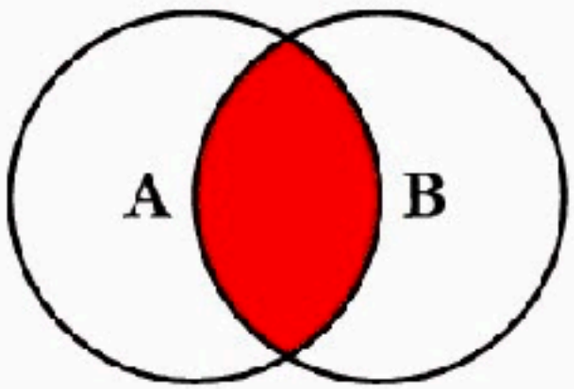
SQL JOINS



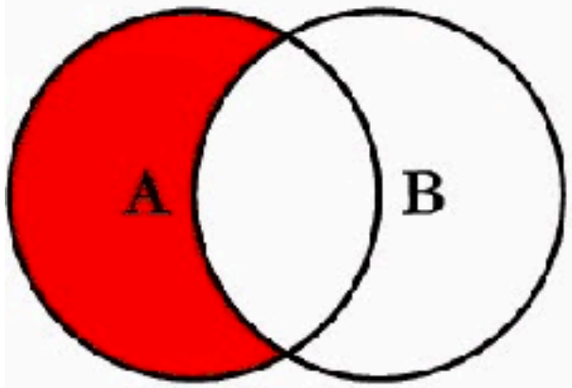

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



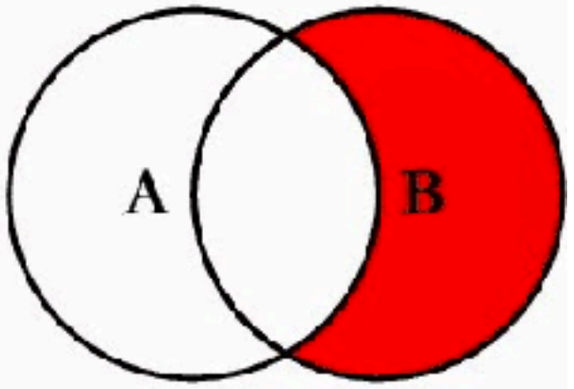

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



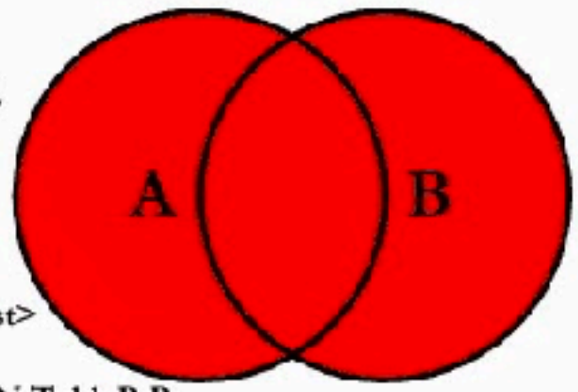

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



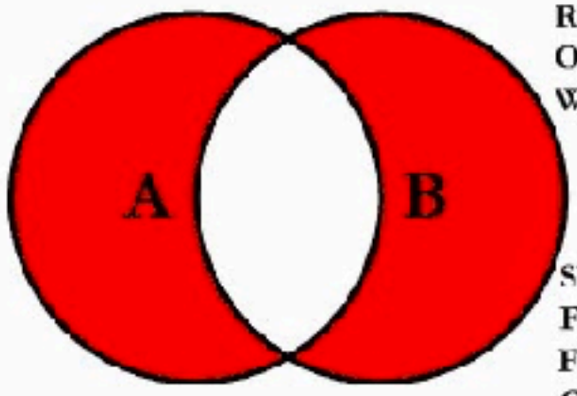

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Don’t forget **INDEXing**

About index ?

Index ?

Why indexes in the first place ?
How do you add an index ?
What tools to help with indexing ?

Why indexes in the first place ?

How do you add an index ?

List of query !!
How to work with tables ?
What is your question ?

Students

ID	First name	Last name	Class
1	A1	B1	6A
2	A2	B2	6A
3	A3	B3	6B
4	A4	B4	6B

Questions ?

Get student by ID
Search for students by first name
List all students in a class

Questions ?

```
CREATE TABLE student(
  id int primary key auto_increment,
  first_name varchar(100),
  last_name varchar(100),
  class varchar(5)
);

INSERT INTO student VALUES(1, "A1", "B1", "6A");
INSERT INTO student VALUES(2, "A2", "B2", "6A");
INSERT INTO student VALUES(3, "A3", "B3", "6B");
INSERT INTO student VALUES(4, "A4", "B4", "6B");
```

Questions ?

```
select * from student where id =1\G;

select * from student where class = '6A'\G;
```

What tools to help with indexing ?

```
EXPLAIN select * from student where id =1\G;

EXPLAIN select * from student where class = '6A'\G;
```

What tools to help with indexing ?

```
CREATE INDEX by_student_class_idx ON student (class);
```