SQL 101

(cos everyone loves a DBA)


Such table.col1, table.col2
So table
Many join
Much group by
Wow

History of RDBMS

Relational **D**ata**B**ase **M**anagement **S**ystem

Invented in 1970 (ish) at IBM

Most popular type of database

Commercial

Open Source

20 years old (first release 1995)

second most widely used RDBMS

most widely used open-source RDBMS

written in C and C++

cross platform

open source (GPL v2)

Core Terms

- Server
- Databases / Schemas
- Tables
- Columns
- Rows
- Fields
- Users
- Permissions & Grants
- Root

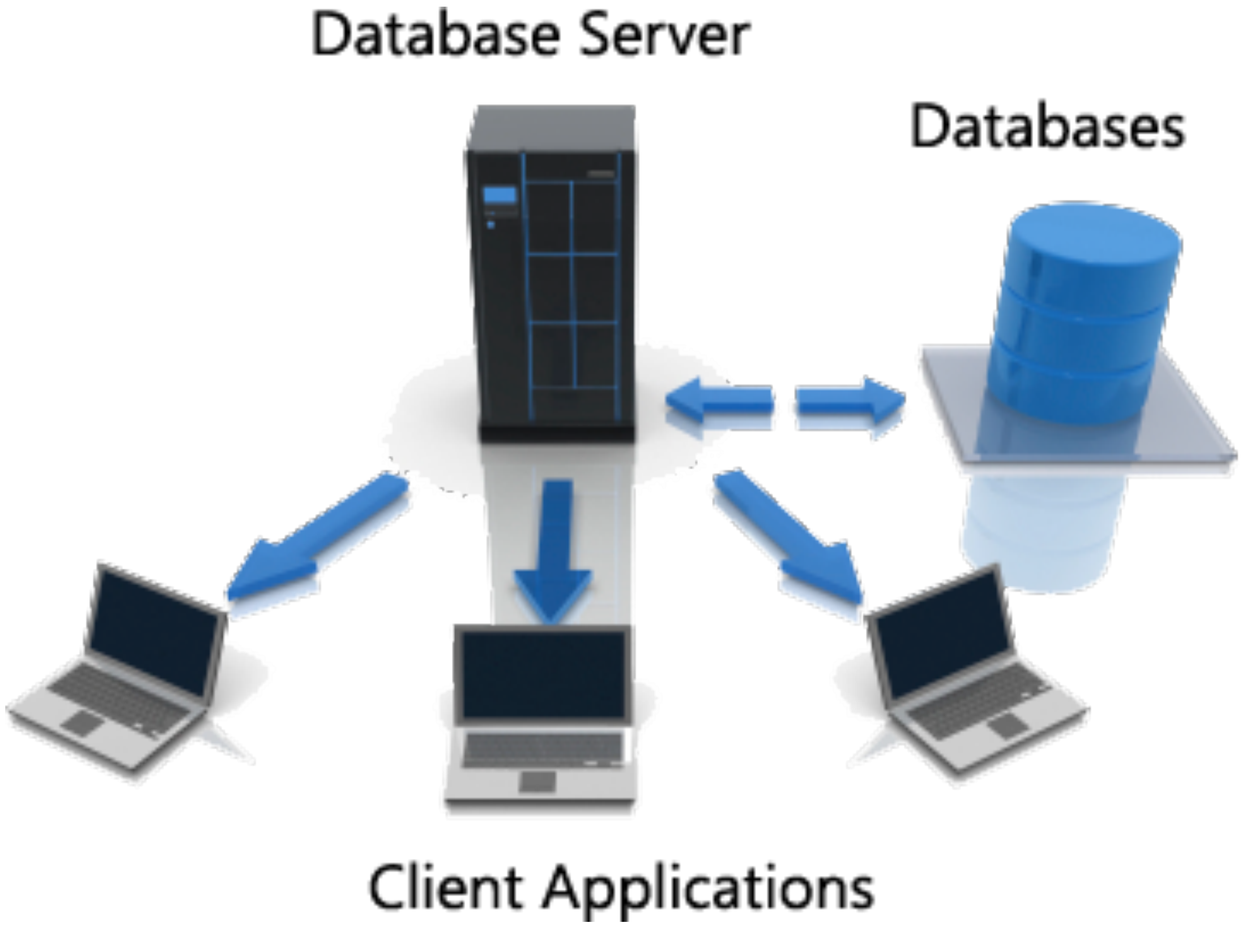
Server & Databases / Schemas

One server has many Databases

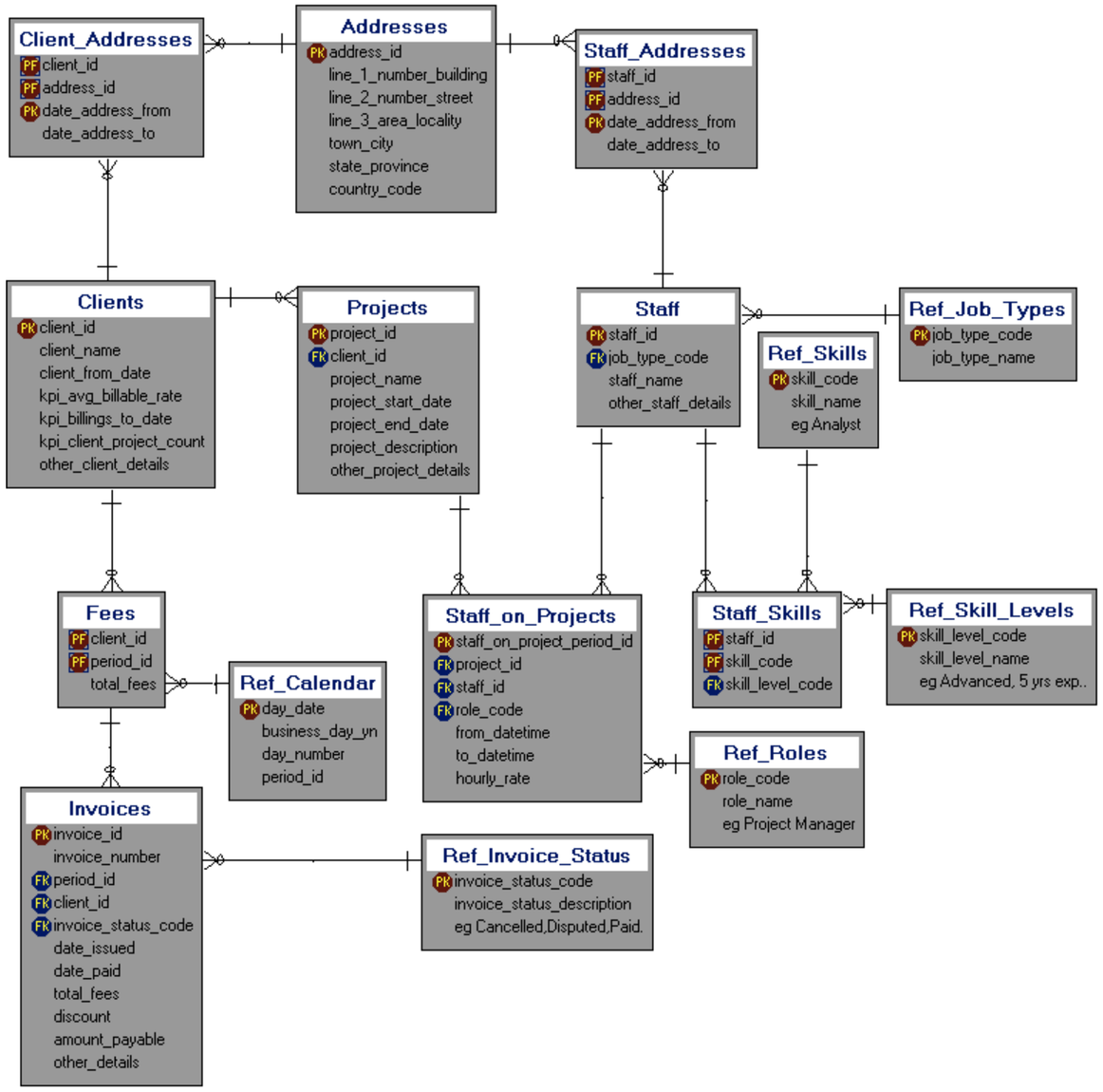
Also known as Schemas

User Schemas (yours)

System Schemas (don’t touch!)



Tables



One Schema has many Tables

Tables are like an Excel Sheet

Columns & Rows

Big Schemas can have 100s of Tables

Columns, Rows & Fields

Each Table has Many Columns

And many Rows

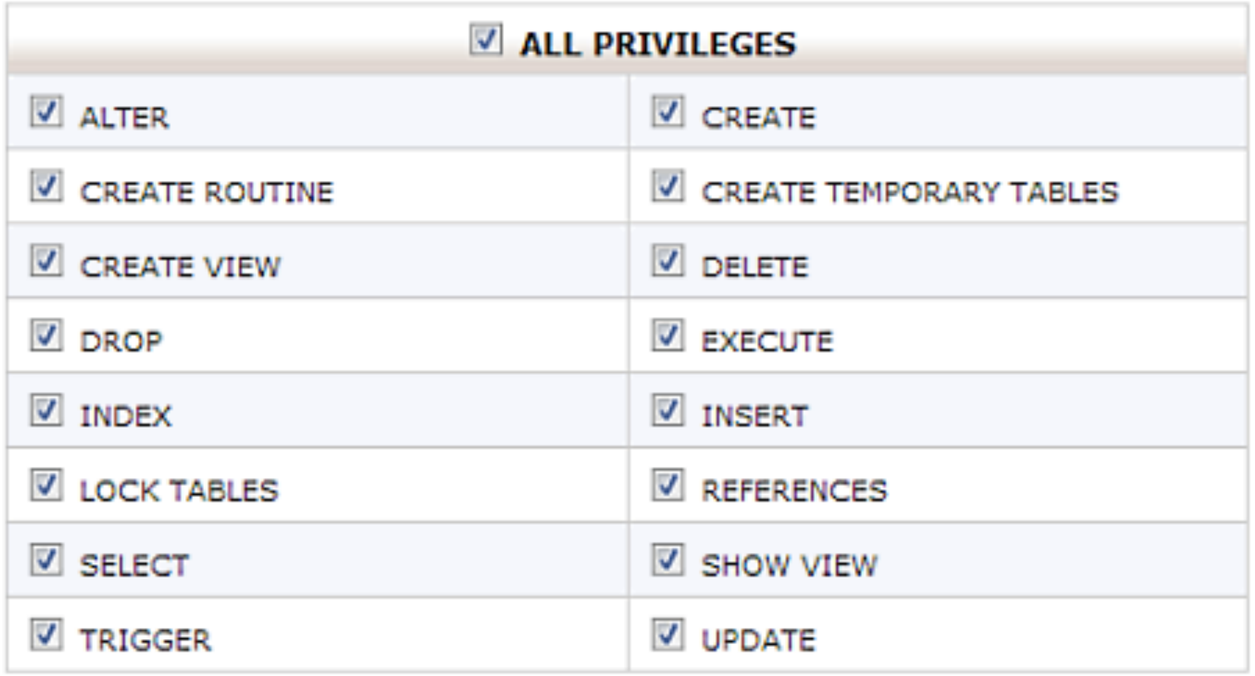
One row per data item

Row & Column is a Field
not a cell! This isn't Excel

Each Column has a specific Type
and other properties

First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

Users & Permissions & Grants



Very granular control over user access

Users can be restricted by:

- Host (IP Address)
- Database
- Table
- Privilege

Root user == god on the system

Do NOT use root!

Data Types

- Text
- Numeric
- Dates & Times
- Nulls
- Auto Increment
- Keys

Data Types - Text

Char: Fixed Length Mixed Chars 123

Varchar: Variable Length Mixed Chars 123

Text: Long blocks of text

Blob: Long blocks of binary data (aka byte strings)

Data Types - Numeric

TinyInt / Int / BigInt: whole integers

Decimal / Numeric: exact decimal values to specified places

Float / Double: 4 / 8 byte precision floating point

Signed Unsigned = can have negative sign or not

Data Types - Integer Types

Type	Storage	Minimum Value	Maximum Value
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

Data Types - Dates & Time

Date: 2015-06-14

Time: 21:32

Year: 2015 (or 15 which is bad!)

Datetime: 2015-06-14 21:32

Timestamp: 2015-06-14 21:32 (UTC)

PUBLIC SERVICE ANNOUNCEMENT:

OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.

THIS IS ***THE*** CORRECT WAY TO WRITE NUMERIC DATES:

2013-02-27

THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:

02/27/2013 02/27/13 27/02/2013 27/02/13

20130227 2013.02.27 27.02.13 27-02-13

27.2.13 2013. II. 27. $27\frac{1}{2}$-13 2013.158904109

MMXIII-II-XXVII MMXIII $\frac{\text{LVII}}{\text{CCCLXV}}$ 1330300800

$((3+3)\times(111+1)-1)\times 3/3-1/3^3$ 2013 Hisssss

10/11011/1101 02/27/20/13 $\overset{2}{0}\overset{3}{1}\overset{1}{2}\overset{4}{3}7_{5}^{}{}_{67}^{}{}_{8}$

Nulls, Auto Increment and Keys

Nulls

An empty value (if the column allows or is nullable)
Not the same as ‘’, 0, false or ‘null’

Auto Increment

An int type that counts up by one each time

Primary Key

The main ID for a table - often an int with auto increment
(also creates an index on the column)

Foreign Key

A column in table that references the Primary Key in another

SQL

Structured **Q**uery **L**angauge

Query
SELECT

Manipulation
INSERT, UPDATE, DELETE

Joins
INNER, LEFT

SQL

Structured **Q**uery **L**angauge

Query
SELECT

Manipulation
INSERT, UPDATE, DELETE

Joins
INNER, LEFT

Anatomy of a SQL Statement / Query

Command	`SELECT`

Query Scope	`first_name,`
	`last_name,`
	`date_of_birth`

Table Selector	`FROM` `people`
Conditions	`WHERE` `home_town = 'St. Helier'`

Anatomy of a SQL Statement / Query

Command	`SELECT`
Query Scope	`*`
Table Selector	`FROM` `people`
Conditions	

Anatomy of a SQL Statement / Query

Command	`UPDATE`
Table Selector	`people`
Query Assignment	`SET` `last_name = 'Crapaud'`
Conditions	`WHERE` `home_town = 'St. Helier'`

Clients

MySQL Workbench
(Cross Platform)

Sequel Pro
(OSX)

Heidi SQL
(Windows)

Let’s play!

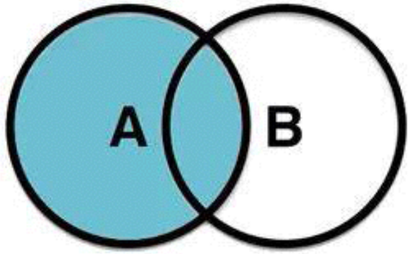
Host: **37.139.7.66**

User: **djlcpstudent**

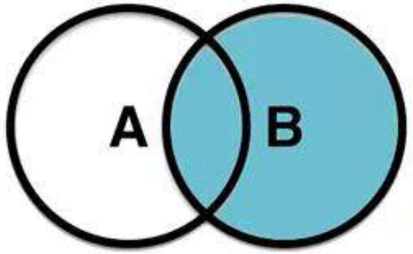
Pass: **SqlStudy.2016**

D/B : **sakila**

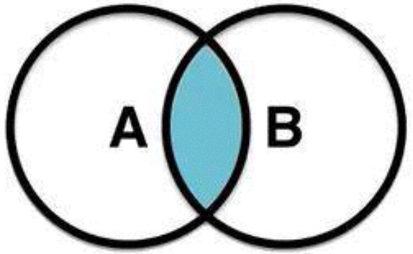
SQL JOINS



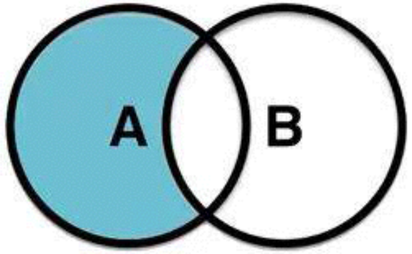

```
SELECT <fields list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



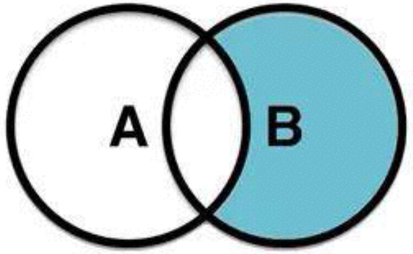

```
SELECT <fields list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



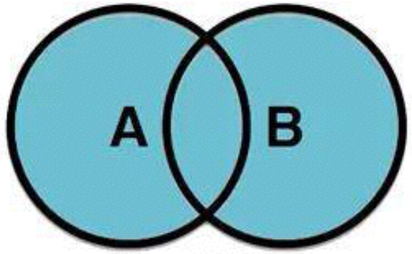

```
SELECT <fields list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



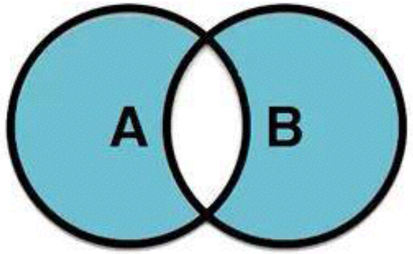

```
SELECT <fields list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



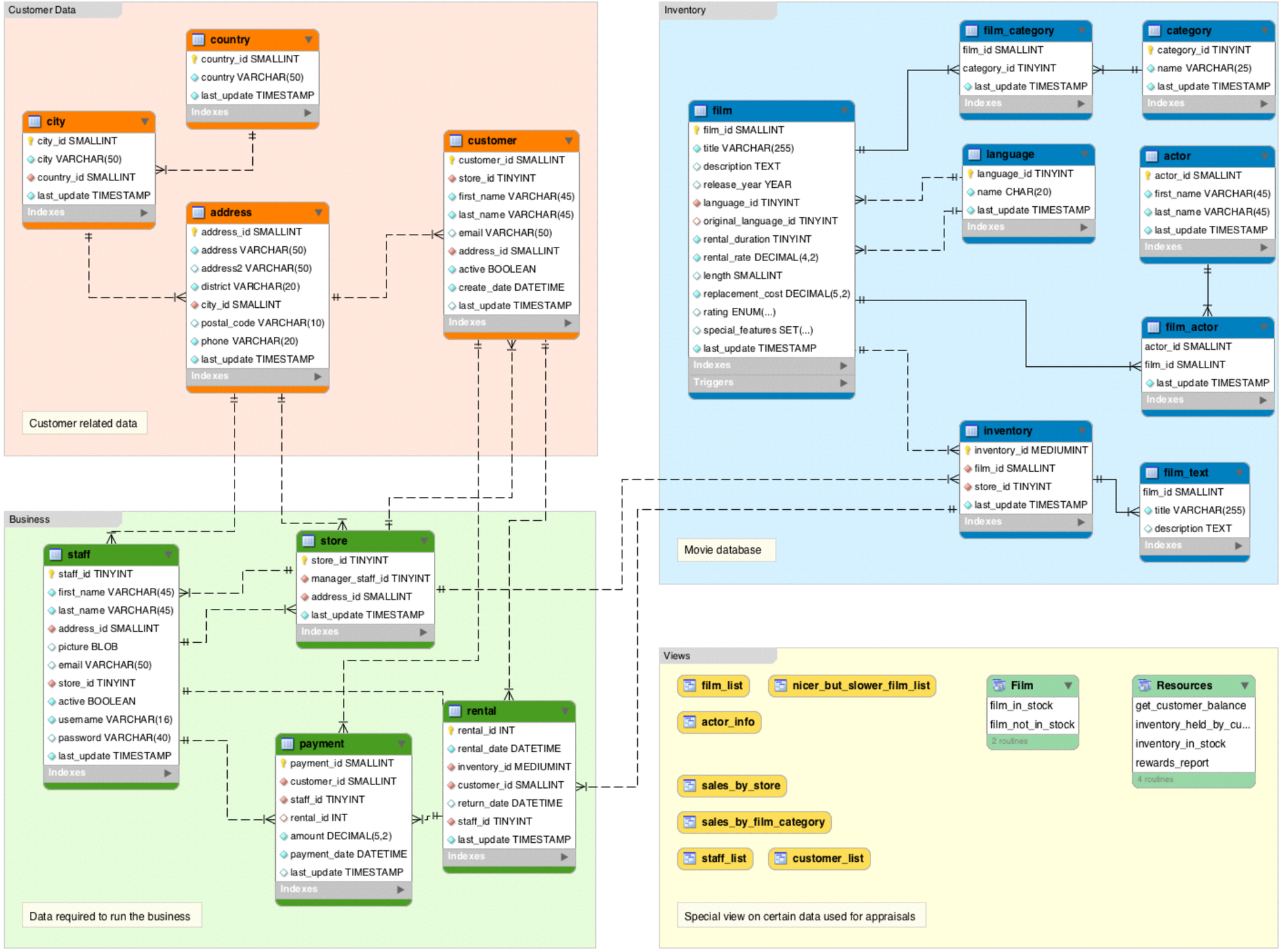

```
SELECT <fields list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




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




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


Customer Data
country
country_id SMALLINT
country VARCHAR(50)
last_update TIMESTAMP
Indexes
city
city_id SMALLINT
city VARCHAR(50)
country_id SMALLINT
last_update TIMESTAMP
Indexes
address
address_id SMALLINT
address VARCHAR(50)
address2 VARCHAR(50)
district VARCHAR(20)
city_id SMALLINT
postal_code VARCHAR(10)
phone VARCHAR(20)
last_update TIMESTAMP
Indexes
customer
customer_id SMALLINT
store_id TINYINT
first_name VARCHAR(45)
last_name VARCHAR(45)
email VARCHAR(50)
address_id SMALLINT
active BOOLEAN
create_date DATETIME
last_update TIMESTAMP
Indexes
Customer related data
Inventory
film_category
film_id SMALLINT
category_id TINYINT
last_update TIMESTAMP
Indexes
category
category_id TINYINT
name VARCHAR(25)
last_update TIMESTAMP
Indexes
film
film_id SMALLINT
title VARCHAR(255)
description TEXT
release_year YEAR
language_id TINYINT
original_language_id TINYINT
rental_duration TINYINT
rental_rate DECIMAL(4,2)
length SMALLINT
replacement_cost DECIMAL(5,2)
rating ENUM(...)
special_features SET(...)
last_update TIMESTAMP
Indexes
Triggers
language
language_id TINYINT
name CHAR(20)
last_update TIMESTAMP
Indexes
actor
actor_id SMALLINT
first_name VARCHAR(45)
last_name VARCHAR(45)
last_update TIMESTAMP
Indexes
film_actor
actor_id SMALLINT
film_id SMALLINT
last_update TIMESTAMP
Indexes
inventory
inventory_id MEDIUMINT
film_id SMALLINT
store_id TINYINT
last_update TIMESTAMP
Indexes
film_text
film_id SMALLINT
title VARCHAR(255)
description TEXT
Indexes
Movie database
Business
staff
staff_id TINYINT
first_name VARCHAR(45)
last_name VARCHAR(45)
address_id SMALLINT
picture BLOB
email VARCHAR(50)
store_id TINYINT
active BOOLEAN
username VARCHAR(16)
password VARCHAR(40)
last_update TIMESTAMP
Indexes
store
store_id TINYINT
manager_staff_id TINYINT
address_id SMALLINT
last_update TIMESTAMP
Indexes
payment
payment_id SMALLINT
customer_id SMALLINT
staff_id TINYINT
rental_id INT
amount DECIMAL(5,2)
payment_date DATETIME
last_update TIMESTAMP
Indexes
rental
rental_id INT
rental_date DATETIME
inventory_id MEDIUMINT
customer_id SMALLINT
return_date DATETIME
staff_id TINYINT
last_update TIMESTAMP
Indexes
Data required to run the business
Views
film_list
nicer_but_slower_film_list
actor_info
Film
film_in_stock
film_not_in_stock
2 routines
Resources
get_customer_balance
inventory_held_by_cu...
inventory_in_stock
rewards_report
4 routines
sales_by_store
sales_by_film_category
staff_list
customer_list
Special view on certain data used for appraisals