Introduction to Databases

Terminology

- **Database**: Organized collection of data
- **Table**: Used to store/ club data same as excel sheet.
- **Row/Tuple**: is an collection of single entity attributes or group of related data.
- **Column**: data of one attribute type.
- **Super Key**: group of single or multiple keys which identifies rows in a table
- **Candidate Key**: A super key with no repeated attribute
- **Primary Key**: Uniquely identify a row (entity) into table. Can not contain NULL.
- **Alternate Key**: Candidate keys which are not primary key

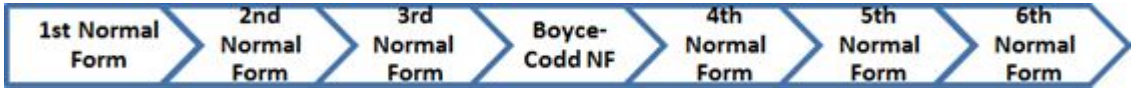
Terminology Contd..

- **Surrogate Key**: Artificial primary key
- **Unique Key**: Unique value among for whole table into same column.
- **Foreign Key**: A link between two tables.
- **Compound (Composite) Key**: is a key that consists of multiple column.
- **Index**: same as book index, used to increase performance of select queries.

https://www.guru99.com/dbms-keys.html

Normalization

Normalization is a concept to break data into multiple table to reduce redundancy of it. It reduce the storage cost of data as well as maintainability also increases.



1st Normal Form

2nd Normal Form

3rd Normal Form

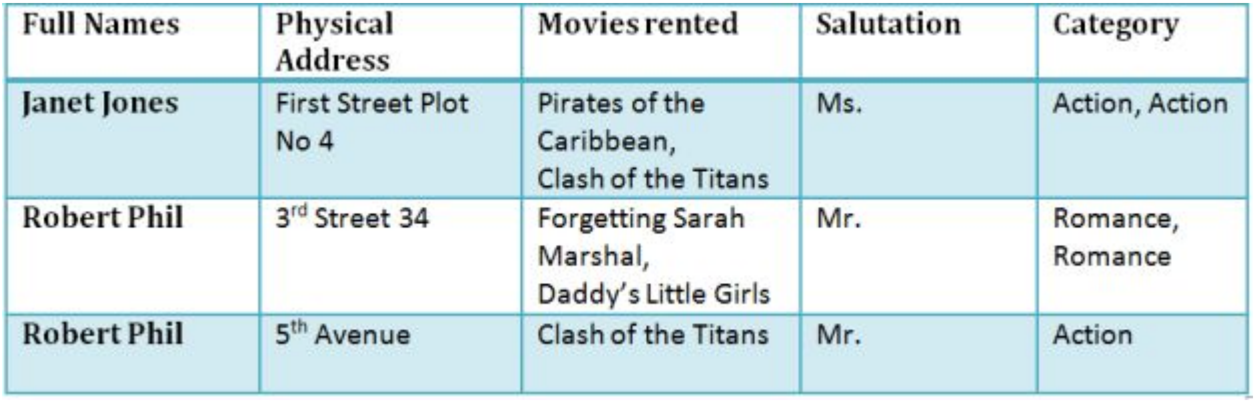
Boyce-Codd NF

4th Normal Form

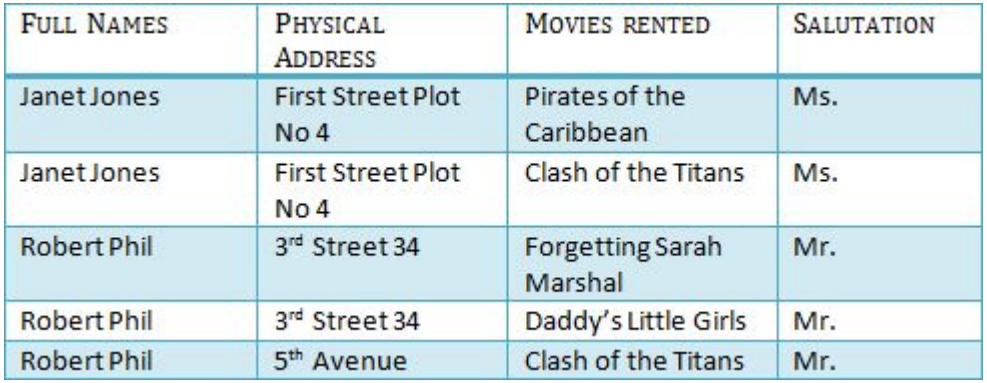
5th Normal Form

6th Normal Form

Normalization Contd.. (1st NF)

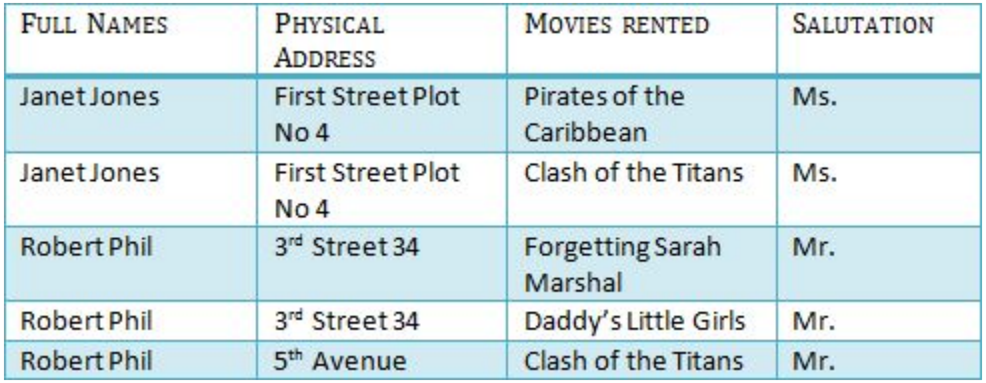


Full Names	Physical Address	Movies rented	Salutation	Category
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.	Action, Action
Robert Phil	3rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.	Romance, Romance
Robert Phil	5th Avenue	Clash of the Titans	Mr.	Action

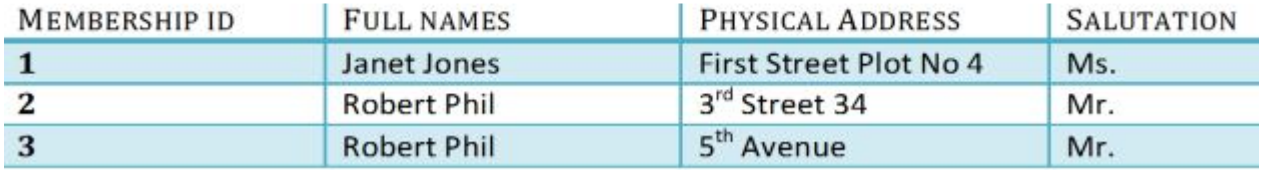


Full Names	Physical Address	Movies rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

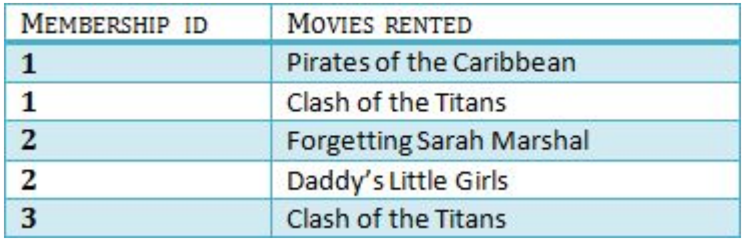
Normalization Contd.. (2nd NF)



Full Names	Physical Address	Movies rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

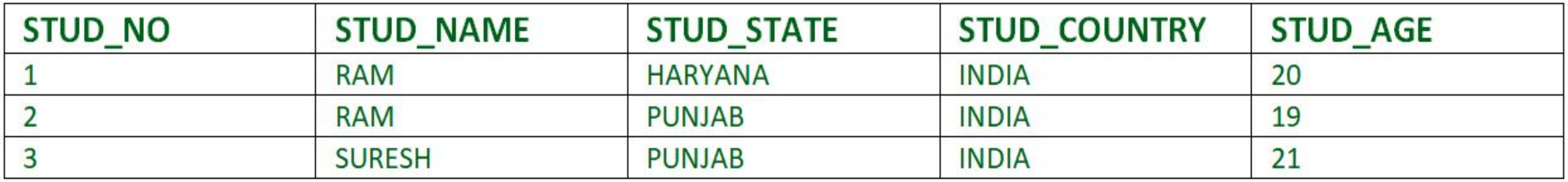


Membership ID	Full Names	Physical Address	Salutation
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.



Membership ID	Movies rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Normalization Contd.. (3rd NF)



STUD_NO	STUD_NAME	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	HARYANA	INDIA	20
2	RAM	PUNJAB	INDIA	19
3	SURESH	PUNJAB	INDIA	21

Table 4

https://www.guru99.com/database-normalization.html

SQL Statement

DDL – Data Definition Language

DDL is used to define the structure that holds the data. For example, Create, Alter, Drop and Truncate table.

DML– Data Manipulation Language

DML is used for manipulation of the data itself. Typical operations are Insert, Delete, Update and retrieving the data from the table. Select statement is considered as a limited version of DML, since it can't change data in the database. But it can perform operations on data retrieved from DBMS, before the results are returned to the calling function.

DCL– Data Control Language

DCL is used to control the visibility of data like granting database access and set privileges to create tables etc. Example - Grant, Revoke access permission to the user to access data in database.

Agenda

- RDBMS Terminology
- Normalization and Type of Sql statement
- Create, delete database & table
- Mysql query, projection, limiting & ordering
- Update & delete query
- Joins
- Indexing – create, delete
- Backup & Restore

Create, drop database

To create database into mysql, you need to execute CREATE command.

eg. **CREATE DATABASE db_name;**

After creating the database you need to select it before using it.
eg. **use db_name;**

To drop database execute the DROP command.

eg. **DROP DATABASE db_name;**

Create/Drop table

To create table into selected database, you need to use CREATE TABLE command.

Syntax: **CREATE TABLE table_name (column_name column_type)**

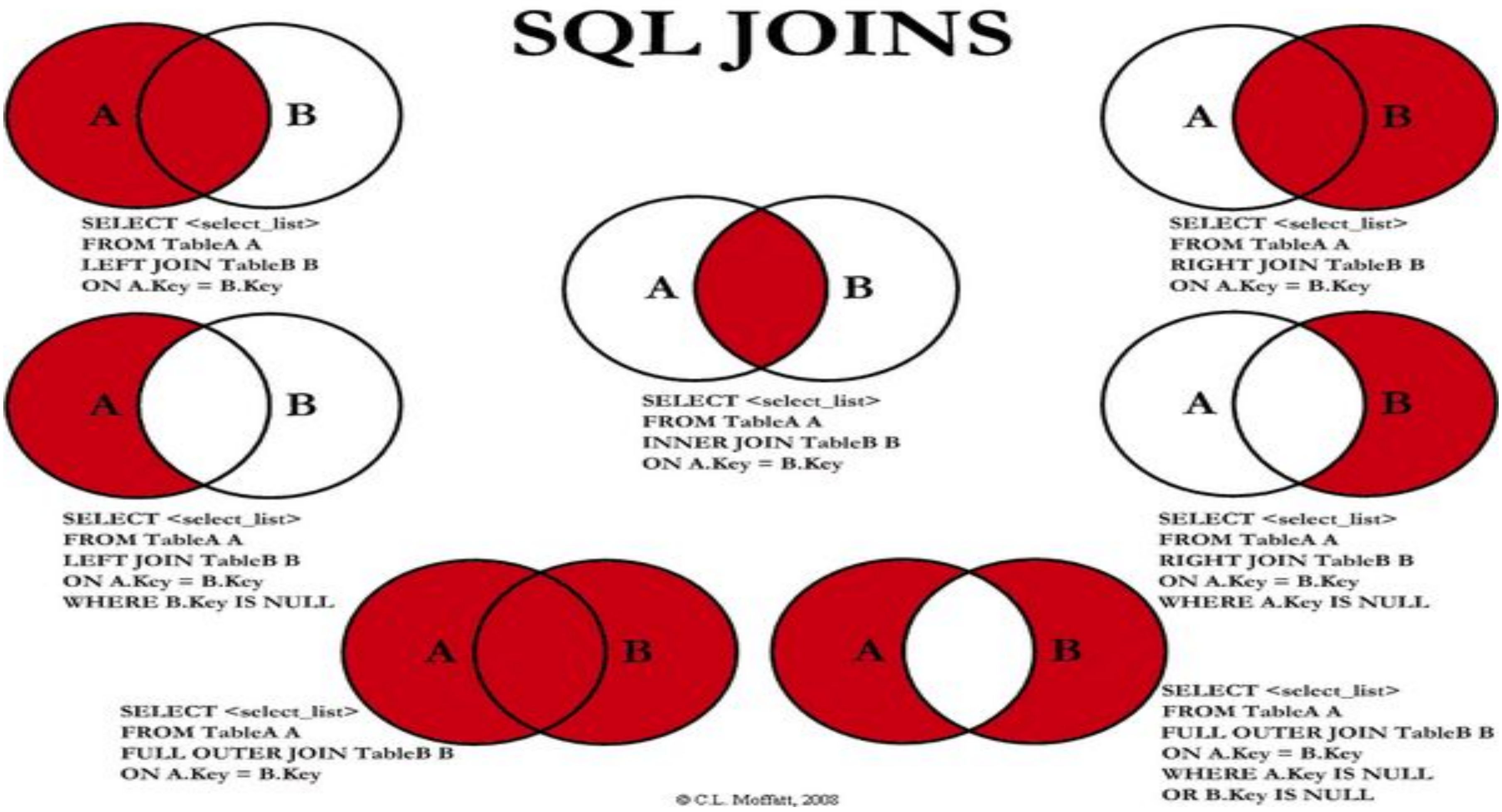
eg.

```
    Create table user (
        id bigint(20) NOT NULL AUTO_INCREMENT,
        email_id varchar(255) NOT NULL,
        is_active bit(1) NOT NULL,
        date_created datetime default NULL,
        primary key (id)
    )
```

To drop a table you need to execute drop table command.

eg. **DROP TABLE table_name;**

Joins Contd..



SQL JOINS

```
SELECT <select_list>
FROM TableA A
LEFT 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
RIGHT 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
```

© C.L. Moffatt, 2008

Insert query

Mysql insert query is used to store data into a table. Syntax:

INSERT INTO table_name (field1, field2, ... field N) values (value1, value2, ... value N)

To insert string values, you need to keep all values either into single quote or double quote.

eg. insert into user (id, email, is_active, date_created) values (1, 'bootcamp@tothenew.com', true, '2016-01-01 06:43:00');

Select query

Mysql select query is used to get data from a table. It returns rows or some aggregated result depending upon the criteria that you have specified.

eg. **Select * from user;**

Above query will return all records of table named with user.

To used filters into select query you need to specify where clause.

Syntax: **SELECT * FROM table_name WHERE condition;**

Where clause

Syntax: SELECT * FROM table_name WHERE clause
eg. select * from user where email_id = “bootcamp@tothenew.com”;

- = Used to check equality
- > Greater than operator
- >= Greater than or equal
- < Less than operator
- <= Less than or equal operator
- BETWEEN .. AND .. Check whether a value is within a range of values
- In() Check whether a value in within a set of values
- IS NOT NULL Not null value test
- IS NULL Used to check null value
- LIKE Used to match pattern from string
- !=, <> Not equal operator
- NOT IN() Opposite of IN operator
- NOT LIKE Opposite of like operator

Multiple operators

In database you can use multiple operators within a single where clause. These operators can be of 2 types.

- **AND**
- **OR**

Multiple operators: AND

AND is used to check multiple conditions in single time. If all specified conditions would be satisfied, then the query would return results.

eg:

```
SELECT *
FROM user
WHERE email_id = “bootcamp@tothenew.com”
AND password = “12345”
```

Multiple operators: OR

OR operator is used to check condition from using multiple operators. Query returns results whenever any one condition is satisfied.

eg:
```
SELECT *
FROM user
WHERE email_id = “bootcamp@tothenew.com”
OR password = “123456”;
```

Multiple operators: AND & OR

You can combine **AND OR** operators into single query to make complex query.

eg:
```
SELECT *
FROM user
WHERE is_active = true
OR (
        email_id = ”bootcamp@tothenew.com“
        AND password = “123456”
);
```

Projection & Limiting Records

By default mysql select query returns all records that are matching criteria. To fetch specific number of records, with specified fields you need to specify projection with limiting records.

Syntax:

```
	SELECT field1, field2 .. fieldN
	FROM table_name
	WHERE clause
	order by field
	LIMIT offset, max
```

eg. **select name from user where is_active = true order by name limit 1, 2**

Above query will return name of 2 active users starting from second position.

NOTE: Here value of offset starts with 0 index.

Update query

Update query is used to update records into a database table. In this query you can update single record as well as multiple records depending upon the criteria specified.

Syntax:

```
UPDATE table_name
SET field1=new_value1, field2=new_value2 .. fieldN=new_valueN
Where clause
```

eg.

```
UPDATE user
SET is_active = false
WHERE email_id = “bootcamp@tothenew.com”
```

Delete Query

Delete query is used to delete specified records from database table. You can delete multiple records from database table by single query.

Syntax: DELETE FROM table_name WHERE clause

eg.

```
	DELETE FROM user
	WHERE email_id = “bootcamp@tothenew.com”
```

If you do not specify email_id in where clause, then it will delete all records from database table. But in this case ID would not reset to 0.

To delete whole table you can use TRUNCATE command also.

TRUNCATE table_name

Joins

Sometimes it is required to get data from multiple tables. To achieve this using multiple queries is not recommended. So MYSQL provide JOIN feature. Join query is used to get data from multiple tables into single query.

You can use JOINS in SELECT, UPDATE as DELETE statements. There are four types of join available into MYSQL.

- Inner Join
- Left Join
- Right Join
- Outer Join

Index

Index in DBMS is used to increase the performance of select query. When you create an index on single or multiple column then a sorted list is maintained into database and query result is returned from that index.

Syntax: ALTER TABLE table_name ADD INDEX index_name (column_list);
eg. **ALTER TABLE user ADD INDEX email_id (email_id);**

NOTE: Primary key and unique key are also part of index. Syntax to create these type of index is explained below:

ALTER TABLE table_name ADD PRIMARY KEY (column_list);
ALTER TABLE table_name ADD UNIQUE KEY (column_list);

Index Contd..

- Compound Index

```
CREATE INDEX my_idx ON my_table(user_id, type);
```

- Drop Index :

```
ALTER TABLE table_name DROP INDEX index_name;
```

- Show indexes:

```
SHOW INDEX FROM table_name;
```

BackUp

You can create backup of whole database into a single file by using following command:

```
mysqldump -uUSERNAME -pPASSWORD DB_NAME > file_name.txt
```

You can also specify table name to take backup of specific table.

To export data/ specific columns into CSV format use following command:

```
SELECT *
FROM table_name
INTO OUTFILE '/tmp/file_name.csv'
FIELDS TERMINATED BY ',' ENCLOSED BY '”'
LINES TERMINATED BY '\r\n';
```

Restore

You can create restore of whole database from a single file using following command:

source filepath/filename.sql

OR

mysql -uUSERNAME -pPASSWORD DB_NAME < FILE_PATH_WITH_NAME

References

MySQL 5.5 Reference Manual :: 3 Tutorial

MySQL Tutorial

Questions?