Introduction to Databases

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
- Back-up & Restore

Terminology

- Database: collection of tables.
- Table: Used to store/ club data same as excel sheet.
- Row: is an collection of single entity attributes or group of related data.
- Column: data of one attribute type.
- Primary Key: Uniquely identify a row (entity) into table. Can not contain NULL.
- Unique Key: Unique value among for whole table into same column. Can contain NULL.

Terminology Contd..

- 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.

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 maintainablity also increases.

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.

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 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)
)

Drop table

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

eg. DROP TABLE table_name;

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 aggegated 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” and password = “12345”;

- = Used to check equality
- > Greater than operator
- >= Used to check database column value is greater than or equal from specified value
- < 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 3 types.
AND
OR

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”

OR Example

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”;

Above query will return all result which are having either email id bootcamp@tothenew.com or password 123456

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

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 specified 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"

NOTE: In the where clause you can pass any combination as explained into previous select query option.

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

Join Query

Sometimes it is required to get data from multiple tables. Ti achieve this using multiple queries is not recommended. So MYSQL provide JOIN feayure. 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

Inner Join

- Inner join is used to get data from multiple tables only where same id is present into both tables.

Syntax: SELECT t1.*, t2.* FROM table_name1 t1 JOIN table_name2 t2 ON t1.id = t2.id

Suppose you have another table user_marks which contains user_id as foreign key and marks of student in each subject.

```
Create table user_marks (
id bigint(20) NOT NULL AUTO_INCREMENT,
user_id bigint(20) NOT NULL,
subject_name varchar(255) NOT NULL,
date_created datetime default NULL,
marks int(10) NOT NULL,
primary key (id)
)
```

Inner Join Contd..

- To get user's information with their marks you need to use INNER JOIN.

eg. SELECT u.*, m.* FROM user u INNER JOIN user_marks m ON u.id =m.user_id;

In the above query you can use where clause, projection and limiting records concept as explained earlier.

NOTE: In RDBMS concepts INNER JOIN is equivalent to c artesian product of two entities.

Left Join

- Left Join is used to get all records which are present into left side table. This join returns all results of inner join plus all records which doesn't have corresponding entry into right side table.

Syntax: SELECT t1.*, t2.* from table_name t1 LEFT JOIN table_name t2 ON t1.id = t2.id

To get all user's details which are either obtained marks into some subject or not.

SELECT u.*, m.* FROM user u LEFT JOIN user_marks m ON u.id = m.user_id;

Right Join

- Right Join is used to get all records which are present into right side table. This join returns all results of inner join plus all records which doesn't have corresponding entry into left side table.

Syntax: SELECT t1.*, t2.* from table_name t1 RIGHT JOIN table_name t2 ON t1.id = t2.id

eg. SELECT u.*, m.* FROM user u RIGHT JOIN user_marks m ON u.id = m.user_id;

When you perform left or right join then null values is printed in case no data available in right or left table.

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);

Index Contd..

ALTER TABLE table_name ADD UNIQUE KEY (column_list);

* 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 Online Documentation
http://dev.mysql.com/doc/refman/5.5/en/tutorial.html
- Onlne Tutorials
http://www.tutorialspoint.com/mysql/

MongoDB

- open-source document database
- Schema Less
- Written in C++
- High performance
- High availability
- Easily scalable

MongoDB

- open-source document database
- Schema Less
- Written in C++
- High performance
- High availability
- Easily scalable

Usecase MongoDB

- Take a example of facebook post which can contains following things
- Post metadata -> Text, written_by, date_time, images
- Likes -> user's name who liked post
- Comments -> username, comment_text, date_time

In RDBMS to manage this you need to create multiple tables & multiple queries while displaying posts. In MongoDB you can achieve this by one collection & one query.

MongoDB Terminology

RDBMS	MongoDB
Database	Database
Table	Collection
Row	Document
column	Field

MongoDB CRUD

Create Database

use DATABASE_NAME

Show databases

show dbs

List Collection

show collections

Insert Document

db.COLLECTION_NAME.insert(document)

List Documents

db.COLLECTION_NAME.find()

Delete Document

db.mycol.remove()

TO THE NEW

Questions?