Day 1

```
* Data, processing, Information
* Database
* Need Of Database
* Applications
* DBMS vs RDBMS
* MySql Introduction
```

Day 2

- Databases do not understand C/C++/Java language. It can understand only SQL("Sequel")[Structured Query Language].
- Initial name of SQL was RQBE(Relational Query By Example). It is introduced by IBM in 1975.
- ANSI is responsible for standardzing SQL. It means that SQL is common for all databases.
- In 2005, code of sql was rewritten in Java.
- Sub Division of SQL commands

1. Data Definition Language[DDL]
   - CREATE
   - ALTER
   - RENAME
   - DESCRIBE
   - DROP
   - TRUNCATE
2. Data Manipulation Language[DML]
   - INSERT
   - UPDATE
   - DELETE
3. Data Query Language[DQL]
   - SELECT
4. Transaction Control Language[TCL]
   - COMMIT
   - ROLLBACK
   - SAVEPOINT
5. Data Control Language[DCL]
   - GRANT
   - REVOKE

Naming Conventions for identifier[Database/ Table / Column name]

1. Maximum 30 characters are allowed
2. Name must begin with character

3. It can contain[A-Z,a-z,0-9,$,_]
4. Reservered word can not be used as identifier.
5. Identifiers are case insensitive.

```
mysql -u root -pmanager
```

```
SELECT user();
SELECT user() FROM DUAL;
```

- DUAL is single row and 2 column table.
- It is a dummy table name, we should use it situations where no tables are referenced:

```
SELECT 2 + 3;
SELECT 2 + 3 FROM DUAL;
```

- Comments

```
-- SELECT 5*5 FROM DUAL;
# SELECT 5*5 FROM DUAL;
/* SELECT 5*5 FROM DUAL; */
```

- CURRENT_USER() is a function which returns user name and host name combination for the MySQL account that the server used to authenticate the current client.
- Check exsting Users

```
SELECT User from user;
```

- Creating New User

```
CREATE USER
'dac'@'localhost'
IDENTIFIED BY
'dac';
```

- Delete User

```
DROP USER 'dac'@'localhost';
```

```
DELETE
FROM user
WHERE
User='dac';
```

- Check User Permissions

```
SHOW GRANTS FOR 'dac'@'localhost';
```

- Assigning permission's to user

```
-- GRANT ALL PRIVILEGES
GRANT ALL
ON
*.*
TO
'dac'@'localhost';
```

```
FLUSH PRIVILEGES;
```

- Removing permission's to user

```
-- REVOKE INSERT ON *.* FROM 'jeffrey'@'localhost';

-- REVOKE ALL PRIVILEGES
REVOKE ALL
ON
*.*
FROM
'dac'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
SHOW GRANTS FOR 'dac'@'localhost';
```

```
-- mysql -u dac -p
mysql -u dac -pdac
```

- In context of SQL, database is also called as schema.
- List databases:

```
SHOW DATABASES;
```

- Create new database

```
-- CREATE DATABASE dac_db;  -- or
CREATE SCHEMA dac_db;
```

- Check currenlty selected database

```
-- SELECT DATABASE() -- or
SELECT DATABASE() FROM DUAL;
```

```
-- SELECT SCHEMA() -- or
SELECT SCHEMA() FROM DUAL;
```

- Working with database

```
USE dac_db;
SELECT DATABASE() FROM DUAL;
```

- List tables from database

```
SHOW TABLES;
```

- Creating Table
- Syntax: CREATE TABLE tbl_name (col_name column_definition);

```
CREATE TABLE books
(
    id INT,
    name VARCHAR(50),
    author VARCHAR(50),
    subject VARCHAR(50),
    price FLOAT
);
```

```
SHOW TABLES;
```

- CHECK TABLE STRUCTURE

```
DESCRIBE books; -- or
DESC books; -- or
EXPLAIN books;  -- or
SHOW COLUMNS FROM books;
```

- Rename Table

```
RENAME TABLE books TO books_tbl;
```

- If we want to modify table structure then we should use ALTER statement.
- How to rename column?
  - Syntax: ALTER TABLE tbl_name RENAME COLUMN old_col_name TO new_col_name;

```
ALTER TABLE books
RENAME COLUMN id TO book_id;

ALTER TABLE books
RENAME COLUMN name TO book_name;

ALTER TABLE books
RENAME COLUMN author TO author_name;

ALTER TABLE books
RENAME COLUMN subject TO subject_name;
```

- How to modify column Definition?
  - Syntax ALTER TABLE tbl_name MODIFY [COLUMN] col_name column_definition.

```
ALTER TABLE books
MODIFY COLUMN book_name VARCHAR(256 );

ALTER TABLE books
MODIFY author_name VARCHAR(256 );

ALTER TABLE books
MODIFY price DOUBLE;

ALTER TABLE books
MODIFY book_id INT(5);
```

- How to add new column in table?
    - Syntax: ALTER TABLE tbl_name ADD [COLUMN] col_name column_definition

```
ALTER TABLE books
ADD COLUMN pub_name VARCHAR(50);
```

- How to drop column from table?
    - Syntax: ALTER TABLE tbl_name DROP [COLUMN] col_name

```
ALTER TABLE books
DROP COLUMN pub_name;
```

- How to insert record into table?
    - Syntax: INSERT INTO tbl_name (col_name1 , col_name2 ...) VALUES (value_list)

```
INSERT INTO books
( book_id, book_name, author_name, subject_name, price )
VALUES
( 1, 'Let Us C', 'Yashwant Kanetkar', 'C', 450 );
```

```
INSERT INTO books
VALUES
( 2, "More Effective C++", "Scott Mayers", "C++", 550 );
```

```
INSERT INTO books
(book_id, subject_name, book_name, author_name, price )
VALUES
( 3, 'Java','Java Certification', 'Khalid Mughal', 650 );
```

```
INSERT INTO books
(book_id, book_name , price )
VALUES
( 4, 'CLR Via C#', 850 );
```

```
INSERT INTO books
VALUES
( 5, 'OS Concepts',null,NULL,500 );
```

```
INSERT INTO books()VALUES( );
```

```
INSERT INTO books
VALUES
(6,'The C Prog Lang.','Dennis Ritche','C', 450),
(7,'C++ Complete Reference','Herbert Schildt', 'C++', 600),
(8,'Java Head First', 'Kathy Siera','Java', 800);
```

- How to view records/rows?
  - SELECT is used to retrieve rows selected from one or more tables
  - Syntax SELECT FROM table_references;

```
SELECT * FROM books;
```

- How to create copy of table?

```
CREATE TABLE new_books
AS
SELECT * FROM books;
```

```
CREATE TABLE new_books_tbl
AS
SELECT book_id, subject_name, book_name, author_name, price FROM books;
```

- How to copy table structure?

```
CREATE TABLE book_table LIKE books;
```

```
INSERT INTO book_table
( SELECT * FROM books );
```

- How to import sql file?

```
CREATE DATABASE classwork;
USE classwork;
```

```
--SOURCE (Drag and Drop ) .sql file here
SOURCE /Users/sandeepkulange/Desktop/DBT/classwork-db.sql;
```

- Fetch records from all columns

```
SELECT
id, name, author, subject, price
FROM
books;
```

```
SELECT
*
FROM
books;
```

- Fetch records from few columns

```
SELECT
name, author, price
FROM
books;
```

```
SELECT
-- name, author, price, price + price * 0.10
-- name, author, price, price + price * 0.10 AS Final_Price
-- name, author, price, price + price * 0.10 AS "Final Price"
-- name, author, price, price + price * 0.10 AS 'Final Price'
-- name, author, price, price + price * 0.10 'Final Price'
name Name, author Author, price Price, price + price * 0.10 'Final Price'
FROM
books;
```

- DISTINCT

```
-- SELECT  author  FROM books;
SELECT DISTINCT author  FROM books;
```

- LIMIT

```
-- SELECT * FROM books LIMIT 4;
SELECT * FROM books LIMIT 4,3;
```

```sql
SELECT
*
FROM books
-- ORDER BY price ASC;
-- ORDER BY price;
ORDER BY price DESC;
```

```sql
SELECT
*
FROM books
-- ORDER BY subject, author;
-- ORDER BY subject ASC, author DESC;
ORDER BY subject DESC, author ASC;
```

```sql
SELECT
*
FROM books
-- ORDER BY subject;
ORDER BY 4;
```

```sql
SELECT
id, name, author, price, price + price * 0.10 Final
FROM books
-- ORDER BY Final;
-- ORDER BY Final DESC;
ORDER BY 5 DESC;
```

- How to delete table?
  - DROP TABLE removes one or more tables.
  - Syntax: DROP TABLE tbl_name;
  - It is a DDL statement, which removes table data as well as table structure.

```sql
DROP TABLE new_books_tbl;
DROP TABLE book_table, new_books;
```

- How to truncate records?
  - TRUNCATE TABLE empties a table completely.
  - Logically, TRUNCATE TABLE is similar to a DELETE statement that deletes all rows, or a sequence of DROP TABLE and CREATE TABLE statements.

- Syntax : TRUNCATE TABLE tbl_name;

```
TRUNCATE TABLE books;
```

→ Which user has logged in?

Keyword ←

- SELECT user(); --OR
- SELECT user FROM DUAL;
- SELECT CURRENT_USER();

DUAL — 1 x 2 dummy table.

Ex. SELECT 2 +3 ; or

SELECT 2x3 FROM DUAL;

→ Comment in SQL:

- -- <Command>
- # <command>
- /* <Command> */

→ To create New User:

- CREATE USER 'dac'@'localhost' IDENTIFIED BY 'dac' ;

↑ user will work on local machine.

→ To drop User:

- DROP USER 'dac'@'localhost'; OR
- DELETE FROM user WHERE User= 'dac';

↳ This Command in mysql Database.

→ To check permission for User:

- SHOW GRANTS FOR 'dac'@'localhost';

Note: By defaul new user don't have any permission.

→ To Assign Permissions to user:

- GRANT

  [ALL] PRIVILEGES → optional

  Database Name ← ON *.* → Table Name

  TO 'dac'@'localhost';

→ To reflect the changes:

- FLUSH PRIVILEGES;

→ To Remove Permissions:

- REVOKE

  [ALL] PRIVILEGES

  ON *.*

  FROM 'dac'@localhost';

→ To reflect changes:

- FLUSH PRIVILEGES

To Save Grant Permission Permanantly.

→ To list/show database:

- SHOW DATABASES;

→ To create new database:

- CREATE DATABASE db_dac; oR
- CREATE SCHEMA db_dac;

→ check currenty Selected database:

- SELECT DATABASE(); --OR
- SELECT DATABASE FROM DUAL) --OR
- SELECT SCHEMA ();

→ To select / use/ Work with database:

- USE db-dac;

→ List tables from database:

- SHOW TABLES.;

→ To create table:

- CREATE TABLE tbl-name (Col-name(s) col-def(s));

↓ Variable ↓ Data type)

- CREAT TABLE Books

```
(    id , INT;             → Variable length string.
     name , VARCHAR,          For fixed length use CHRA)
     author VARCHAR,
     subject VARCHAR,
     price FLOAT
  );
```

→ To check description of table:

- DESCRIBE TABLES; OR

- EXPLAIN books;
- SHOW COLUMN OF Books;
- DESC books;

latin1 - ASCII

Utf8 - UNICODE

→ Rename table:

- RENAME TABLE books TO books_tbl;

Note: doesn't work for database rename:

→ To modify table structure:

→ To rename column:

- ALTER TABLE tbl_name
  RENAME COLUMN old_col_name TO new_clo_nam

→ To modify column defination:

- ALTER TABLE tbl_name
  MODIFY [COLUMN] col_name col_defination;

→ To add new column:

- ALTER TABLE tbl_name
- ADD COLUMN Publisher_name VARCHAR(50);

→ To drop column from table:

- ALTER TABLE tbl_name
  DROP [COLUMN] col_name;

Note: changes done by ALTER command can't be undone.

∴ ROLLBACK(undo) → DDL ✗
→ DML ✓

→ To insert record into the table:

- INSERT INTO tbl-name
  (col-name1, col-name2 .....) → optional → for ordered values + ALL values
  → Required → for un-ordered values & Partial values.
  VALUES
  (val1, Val2....);

→ To view rows/records in a table:

- SELECT * FROM tbl-name;

Which & How many column want to read? → *

Note: In SQL, NULL represents unknown value/ undefined value.

- Size of NULL → 1 Byte.

- INSERT INTO () VALUES(); → To insert NULL Record.

→ Copy Paste a table:

- CREATE TABLE new_tbl
  AS SELECT * ← ALL Column (Can Select Particular Column)
  book_tbl ;

→ create new table having same structure of another table (but with No data):

- CREATE TABLE new_tbl
  LIKE books_tbl;

→ To copy rows from one. ~~rows~~ table into another!

- INSERT INTO book_table
  (SELECT * FROM books);

→ To import Sql file:

- SOURCE filepath;

→ Records Return by SELECT Query is called as "Result set".

rank h

→ Virtual/ Computed/ Derived Column :-

If we generate new column in result set by some business logic then it is virtual Vogi Column

- SELECT

name, author, Price, $\underbrace{\text{price+Price* 0.10}}_{\text{Virtual column.}}$

FROM books;

- SELECT

→ optional

name, author, Price, Price+ Price * 0.10 $\boxed{\text{AS}}$

Alias

" final Price" → --OR, final_Price

FROM books; --OR ' final Price'.

OR

→ • —" —

name NAME, author As AUTHOR ....

—" —

→ To get unique (authors) data from column:

- SELECT DISTINCT authors FROM books;

  → Ignore duplicate data.

→ To select particular rows No. of rows from start:

- SELECT * FROM books LIMIT 5;

  will give first five row.

→ To get particular No. of rows inbetween:

- SELECT * FROM books LIMTT 4,3;

→ sorting Records:

- SELECT * FROM books
  ORDER BY Price ASC;

  → By default it sort by ASC.

  → -- DESC;

  → Here we can give column No. also. (ORDER BY 4)

→ Sorting by 2 column:

— · SELECT * FROM books
ORDER BY subject , author ;

1st 2nd

subject DESC , author ASC ;

→ To Drop table/ Remove table:

- DROP TABLE tbl-name;

Note:- Remove table data as well as table structure.

→ TRUNCATE :- Remove table data but keep table structure.

- TRUNCATE TABLE books;

Note:	TRUNCATE TABLE	Vs DROP TABLE.
Deletes →	only Data	Data + structure

⇒ Datatype Reading Assignment :