# Agneda

- WHERE clause
  - LIKE operator
  - IN operator
  - BETWEEN operator
  - ~~REGEXP operator~~
- DELETE, TRUNCATE & DROP
- UPDATE
- ~~DUAL table~~
- ~~SQL Functions~~

```
HELP SELECT;

SELECT columns FROM table_name
WHERE condition
ORDER BY column_expr
LIMIT n;
```

- String Size = Number of chars * Size of each char.
  - CHAR(10) = 10 * sizeof(char)
    - ASCII encoding --> 10 * 1 = 10 bytes
    - UTF-16 encoding --> 10 * 2 = 20 bytes

## SELECT Query

### WHERE clause

**BETWEEN operator**

- BETWEEN operator
  - WHERE column BETWEEN lower AND upper;
  - Range is inclusive of both ends.
- NOT BETWEEN -> reverse of BETWEEN

```
SELECT USER(), DATABASE();

-- find all emps having salary between 1500 to 3000 -- 1500 >= sal >= 3000.
SELECT * FROM emp WHERE sal >= 1500 AND sal <= 3000;

SELECT * FROM emp WHERE sal BETWEEN 1500 AND 3000;

-- find all emps hired in 1982.
SELECT * FROM emp WHERE hire >= '1982-01-01' AND hire <= '1982-12-31';

SELECT * FROM emp WHERE hire BETWEEN '1982-01-01' AND '1982-12-31';
```

```
-- find all emps having names between C and M.
SELECT * FROM emp WHERE ename BETWEEN 'C' AND 'M';
--| C      |
--| CLARK  |
--| FORD   |
--| JAMES  |
--| JONES  |
--| KING   |
--| M      |
--| MARTIN |
--| MILLER |
--| MZ     |
--| N      |

SELECT * FROM emp WHERE ename BETWEEN 'C' AND 'N';

SELECT * FROM emp WHERE ename BETWEEN 'C' AND 'N' AND ename != 'N';

SELECT * FROM emp WHERE ename BETWEEN 'C' AND 'MZ';

-- find all emps having salary not between 1500 to 3000.
SELECT * FROM emp WHERE sal NOT BETWEEN 1500 AND 3000;
```

### IN operator

- IN operator check for "equality" with multiple values.
  - WHERE column IN (value1, value2, ...)
- NOT IN -- inverse of IN.

```
-- find all ANALYST, MANAGER and PRESIDENT.
SELECT * FROM emp WHERE job = 'ANALYST' OR job = 'MANAGER' OR job = 'PRESIDENT';

SELECT * FROM emp WHERE job IN ('ANALYST', 'MANAGER', 'PRESIDENT');

-- find all emps in dept 10 & 20.
SELECT * FROM emp WHERE deptno = 10 OR deptno = 20;

SELECT * FROM emp WHERE deptno IN (10, 20);

-- find emps JAMES, KING or MARTIN.
SELECT * FROM emp WHERE ename IN ('JAMES', 'KING', 'MARTIN');

-- find all emps not in dept 10 & 20.
SELECT * FROM emp WHERE deptno NOT IN (10, 20);

-- find all emps in dept 20 & 30 having sal in range 2000 to 2500.
SELECT * FROM emp
WHERE deptno IN (20, 30)
AND sal BETWEEN 2000 AND 2500;
```

#### LIKE operator

- In MySQL, String comparison case insensitive.
  - So 'JAMES' and 'James' are same.
- Comparing strings using wild-card chars (patterns).
  - % --> any number of any characters
  - _ --> single occurrence of any character

```
-- find all emps whose name start with 'S'.
SELECT * FROM emp WHERE ename LIKE 'S%';

-- find all emps whose name ends with 'D'.
SELECT * FROM emp WHERE ename LIKE '%D';

-- find all emps whose name contains 'IT'.
SELECT * FROM emp WHERE ename LIKE '%IT%';

-- find all emps whose name is of 4 letters.
SELECT * FROM emp WHERE ename LIKE '____';

-- find all emps whose second letter in 'A'.
SELECT * FROM emp WHERE ename LIKE '_A%';

-- find all emps whose name contains 'A' more than once.
SELECT * FROM emp WHERE ename LIKE '%A%A%';

-- find all emps whose name is in range 'T' to 'Z'.
SELECT * FROM emp
WHERE ename BETWEEN 'T' AND 'Z'
OR ename LIKE 'Z%';
-- BETWEEN T AND Z --> all names between T & Z, but not names Z onwards.
-- ename LIKE Z% --> all names starting with Z
-- OR combines both conditions.

-- find all emps whose name doesn't have I.
SELECT * FROM emp WHERE ename NOT LIKE '%I%';

-- find all emp with 4 letter name but start with A.
SELECT * FROM emp WHERE ename LIKE 'A___';
```

## UPDATE query

- Update the row(s) and column(s) "values".
  - Cannot edit column name or type.
- UPDATE table_name SET column_name=new_value WHERE condition;

```
sunbeam> SELECT * FROM people;
+--------------------+--------+------------+
| name               | gender | birth      |
+--------------------+--------+------------+
| James Bond         | M      | 1960-02-23 |
| Superman           | M      | 1980-04-21 |
| Spiderman          | M      | 1984-12-01 |
| Batman             | M      | 1979-05-20 |
| John Galt          | M      | NULL       |
| Dagny Tagort       | F      | NULL       |
| Fransisco Dankonia | M      | NULL       |
+--------------------+--------+------------+
```

```
SELECT * FROM people;

UPDATE people SET birth='1960-01-01' WHERE name = 'John Galt';

SELECT * FROM people;

UPDATE people SET birth='1964-12-12' WHERE gender = 'F';

SELECT * FROM people;

UPDATE people SET gender='M', birth='1950-12-21';

SELECT * FROM people;
```

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

-- increase price of all books by 10%, if book name contains PROGRAMMING.
UPDATE books SET price = price + price * 0.10
WHERE name LIKE '%PROGRAMMING%';

-- decrease price of all C books by 5%.
 UPDATE books SET price = price - price * 0.05
 WHERE subject = 'C Programming';

-- set price of Herbert Schildt's book of Java to 1000.
UPDATE books SET price = 1000
WHERE author = 'Herbert Schildt' AND subject = 'Java Programming';
```

# DELETE

- DELETE FROM table_name WHERE condition;

```
-- delete John Galt from people.
DELETE FROM people WHERE name = 'John Galt';

SELECT * FROM people;
```

```
-- delete all people whose name ends with 'man'
DELETE FROM people WHERE name LIKE '%man';

SELECT * FROM people;

-- delete all people
DELETE FROM people;

SELECT * FROM people;

DESCRIBE people;
```

- Q: DELETE * FROM tablename;
  - Error.
- Q: Why * is not used in DELETE?
  - "*" means all columns.
  - DELETE will delete the rows (not columns).
- Q. What happens in background when DELETE?

```
TRUNCATE TABLE students;

SELECT * FROM students;

DESCRIBE students;
```

```
DROP TABLE students;
```