Table Queries

Alter (to change the schema)

ADD Column

```
ALTER TABLE table_name
ADD COLUMN column_name datatype constraint;
```

DROP Column

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

RENAME Table

```
ALTER TABLE table_name
RENAME TO new_table_name;
```

Table Queries

Alter (to change the schema)

CHANGE Column (rename)

```
ALTER TABLE table_name
CHANGE COLUMN old_name new_name new_datatype new_constraint;
```

MODIFY Column (modify datatype/ constraint)

```
ALTER TABLE table_name
MODIFY col_name new_datatype new_constraint;
```

```
SET SQL_SAFE_UPDATES = 0;
```

Table Queries

Update (to update existing rows)

```
UPDATE table_name
SET col1 = val1, col2 = val2
WHERE condition;
```

General Order

SELECT *column(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column(s)*
HAVING *condition*
ORDER BY *column(s)* **ASC;**

Having Clause

Similar to Where i.e. applies some condition on rows.
But it is used when we want to apply any condition after grouping.

```
SELECT col1, col2
FROM table_name
GROUP BY col_name(s)
HAVING condition;
```

- WHERE is for the table, HAVING is for a group
- Grouping is necessary for HAVING

Group by Clause

Groups rows that have the same values into summary rows.
It collects data from multiple records and groups the result by one or mor

```
SELECT col1, col2
FROM table_name
GROUP BY col_name(s);
```

*Generally we use group by with some *aggregation function*.

Aggregate Functions

Aggregate functions perform a calculation on a set of values, and return

- COUNT( )
- MAX( )
- MIN( )
- SUM( )
- AVG( )

Example :

```
SELECT max(marks)
FROM student;
```

Order by Clause

To sort in ascending (ASC) or descending order (DESC)

SELECT *col1, col2* **FROM** *table_name*
ORDER BY *col_name(s)* **ASC;**

Where Clause

Frequently used Operators

AND (to check for both conditions to be true)

OR (to check for one of the conditions to be true)

BETWEEN (selects for a given range)

IN (matches any value in the list)

NOT (to negate the given condition)

Constraints

Rules for data in the table

NOT NULL	columns cannot have a null value
UNIQUE	all values in column are different
DEFAULT	sets the default value of a column
CHECK	it can limit the values allowed in a column

Data Types

DATATYPE	DESCRIPTION	USAGE
CHAR	string(0-255), can store characters of fixed length	CHAR(50)
VARCHAR	string(0-255), can store characters up to given length	VARCHAR(50)
BLOB	string(0-65535), can store binary large object	BLOB(1000)
INT	integer(-2,147,483,648 to 2,147,483,647)	INT
TINYINT	integer(-128 to 127)	TINYINT
BIGINT	integer(-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)	BIGINT
BIT	can store x-bit values. x can range from 1 to 64	BIT(2)
FLOAT	Decimal number - with precision to 23 digits	FLOAT
DOUBLE	Decimal number - with 24 to 53 digits	DOUBLE
BOOLEAN	Boolean values 0 or 1	BOOLEAN
DATE	date in format of YYYY-MM-DD ranging from 1000-01-01 to 9999-12-31	DATE
YEAR	year in 4 digits format ranging from 1901 to 2155	YEAR

Table Queries

- Create
- Insert
- Update
- Alter
- Truncate 
- Delete

Database Queries

```
CREATE DATABASE db_name;

CREATE DATABASE IF NOT EXISTS db_name;


DROP DATABASE db_name;

DROP DATABASE IF EXISTS db_name;


SHOW DATABASES;

SHOW TABLES;
```

SQl v/s NoSQL

SQL

Relational Database
(data stored in Tables)

eg - MySQL, Oracle,
PostgreSQL etc.

NoSQL

Non Relational Database
(data stored in document/ke

eg - MongoDb, Cassandra,
Neo4j etc.

Table Queries

Truncate (to delete table's data)

```
TRUNCATE TABLE table_name ;
```