SELECT \*

FROM table\_name

WHERE condition;

{MIN,MAX,SUM,AVG : Aggregate Functions}

SELECT COUNT(column\_name)

FROM table\_name

WHERE condition;

{ % represents 0/1/multiple characters. The underscore sign \_ represents one, single character}

SELECT column1

FROM table\_name

WHERE columnN LIKE pattern;

SELECT \*

FROM Customers

WHERE Country IN ('Germany', 'France', 'UK');

SELECT \*

FROM Products

WHERE Price BETWEEN 10 AND 20;

SELECT \*

FROM Products

ORDER BY Price

DESC;

SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country;

{WHERE keyword cannot be used with aggregate functions}

SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country

HAVING COUNT(CustomerID) > 5;

OPTIMIZE TABLE table\_name;

CREATE INDEX index\_label ON table\_name(column\_name)

CREATE TABLE Persons (

Personid int NOT NULL AUTO\_INCREMENT,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

PRIMARY KEY (Personid)

);

INSERT INTO Persons (FirstName,LastName)

VALUES ('Lars','Monsen');

{To insert a new record into the "Persons" table, we will NOT have to specify a value for the "Personid" column (a unique value will be added automatically)}

ALTER TABLE table\_name PARTITION BY RANGE (ID) (

PARTITION p0 VALUES LESS THAN (4),

PARTITION p1 VALUES LESS THAN (8),

PARTITION p2 VALUES LESS THAN (12),

PARTITION p3 VALUES LESS THAN MAXVALUE

);

ALTER TABLE table\_name PARTITION BY LIST (ID) (

PARTITION p0 VALUES IN (1,2,3,4),

PARTITION p1 VALUES IN (5,6,7,8),

PARTITION p2 VALUES IN (9,10,11,12),

PARTITION p3 VALUES IN (13,14,15,16)

);

ALTER TABLE table\_name PARTITION BY HASH (ID)

PARTITIONS 4;

ALTER TABLE table\_name PARTITION BY KEY ()

PARTITIONS 4;