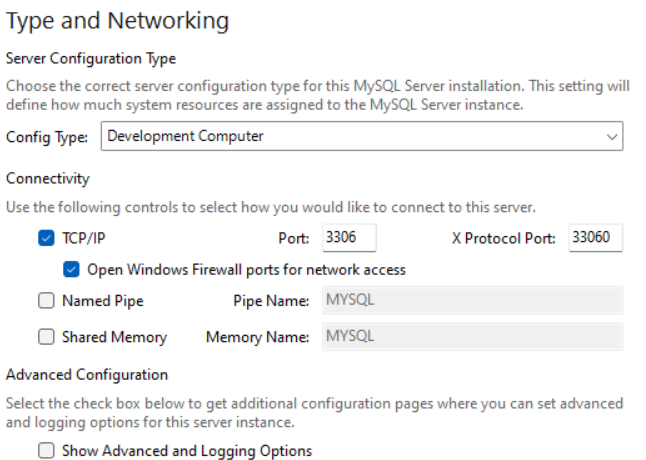
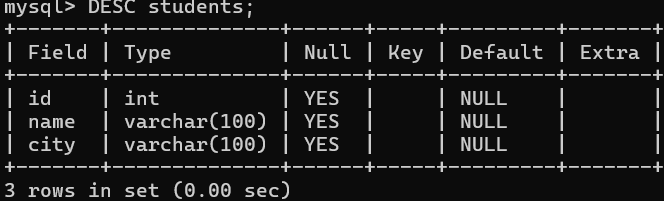
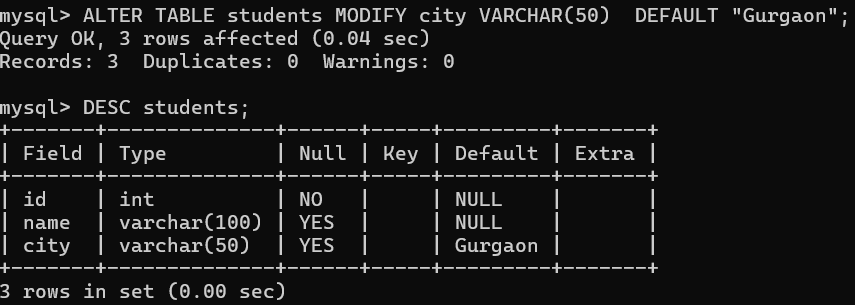
**DataBase:**

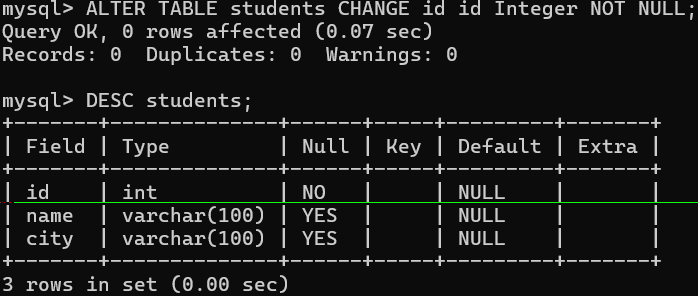
### What is the major difference between MySQL and SQL?

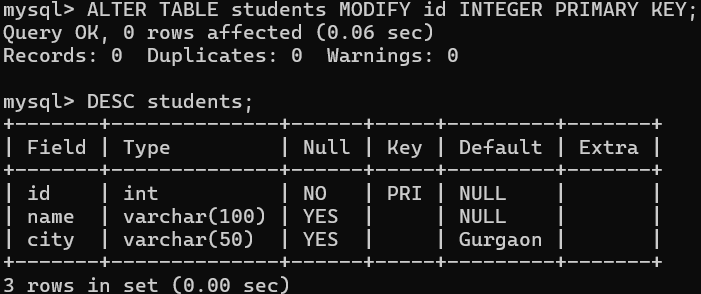
SQL is a query programming language for managing RDBMS. In contrast, MySQL is an RDBMS (Relational Database Management System) that employs SQL. So, the major difference between the two is that MySQL is software, but SQL is a database language.

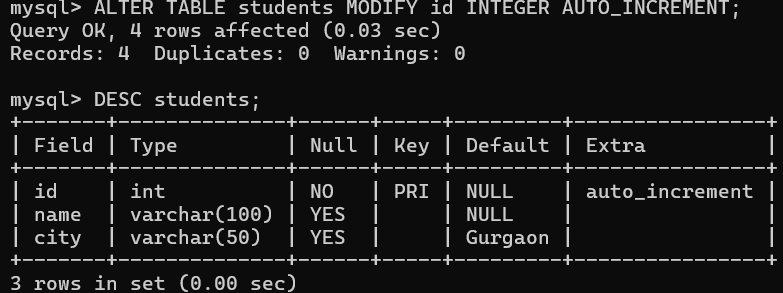


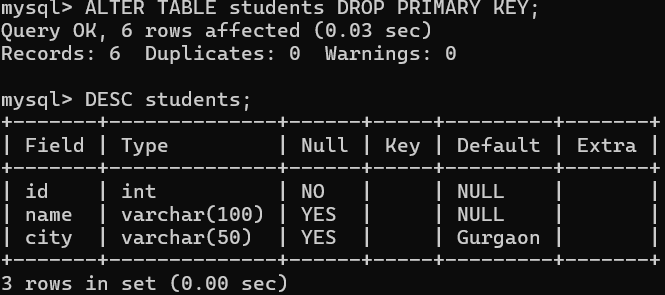


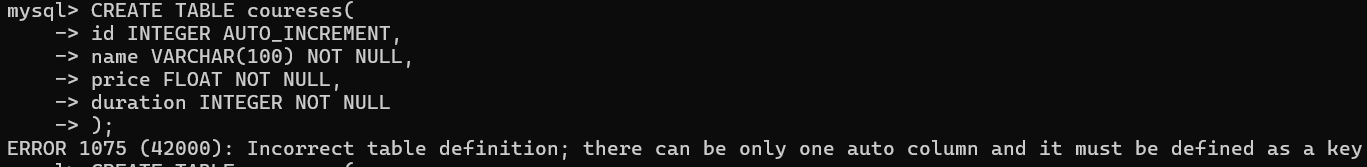




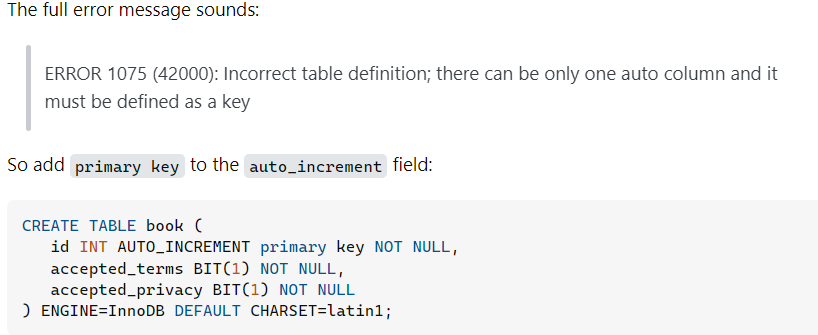


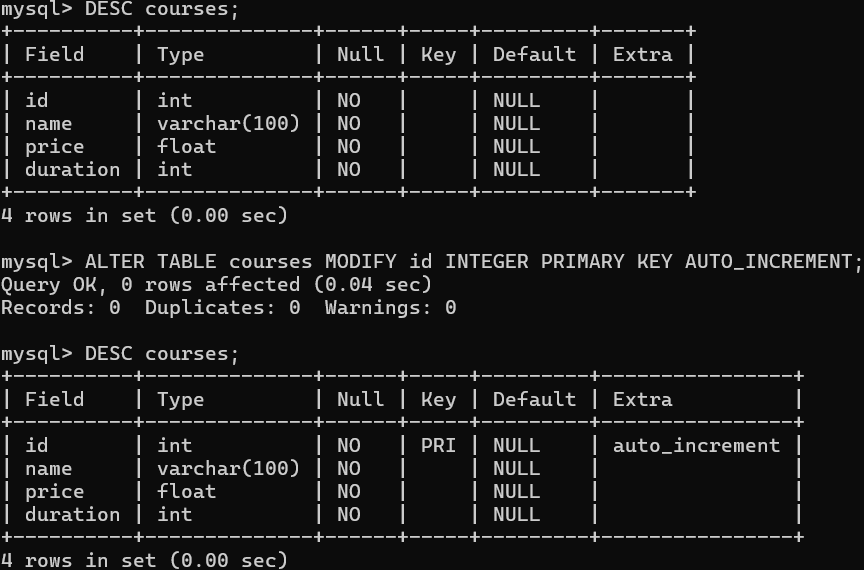






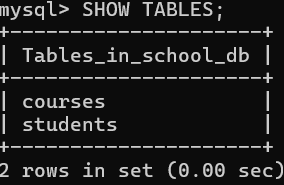
Throwing error, how to fix above error?





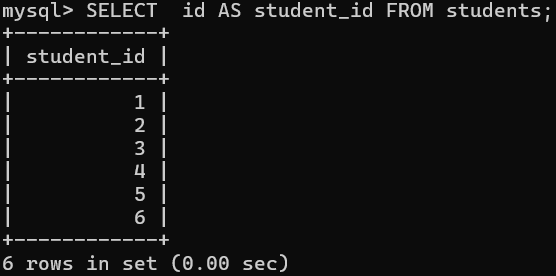
To get all table present for a db:

USE db\_name;

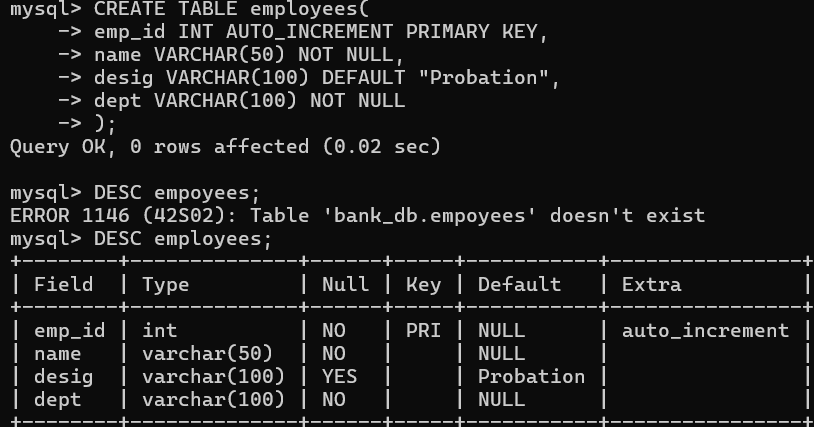


NOTE:

**You cant use AUTO\_INCREMENT without PRIMARY KEY in mysql.**

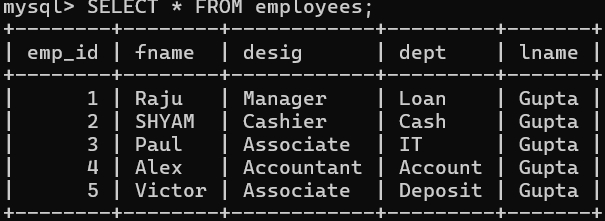


Here in above image we can set alias name of column name “id” as “student\_id”.

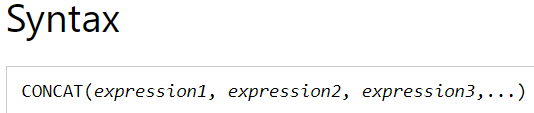


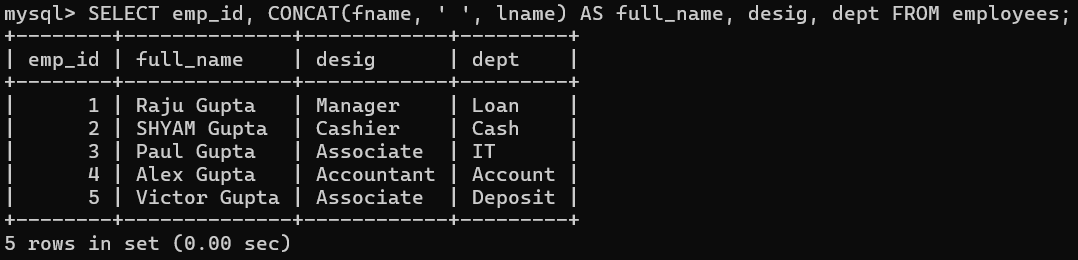
String function in MySQL:

Reference Table: employees:

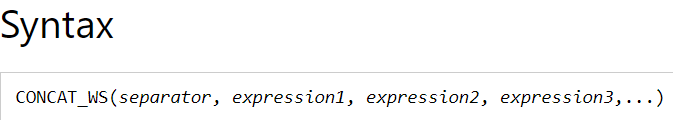


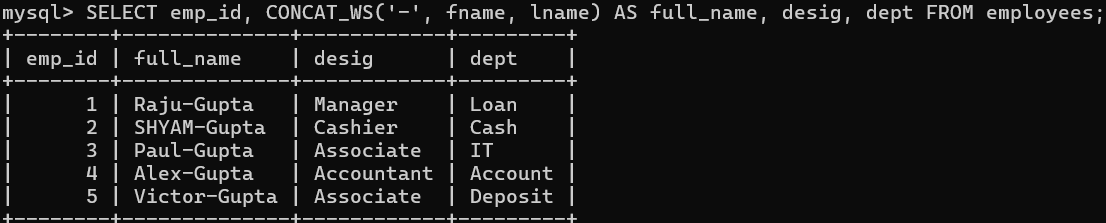
CONCAT:





CONCAT\_WS:





SUBSTRING

REPLACE

LEFT

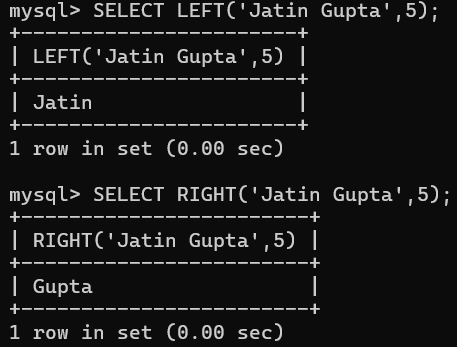
RIGHT

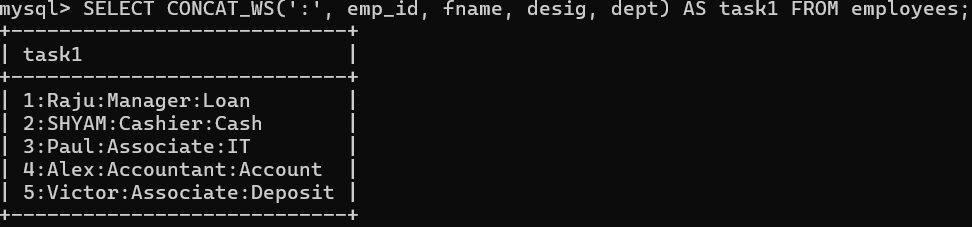
REVERSE

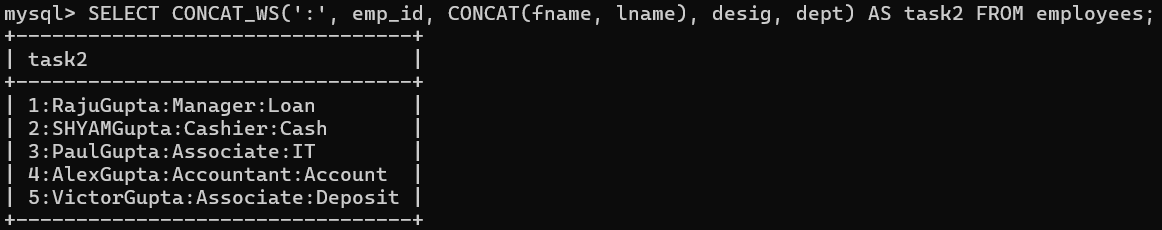
REPEAT

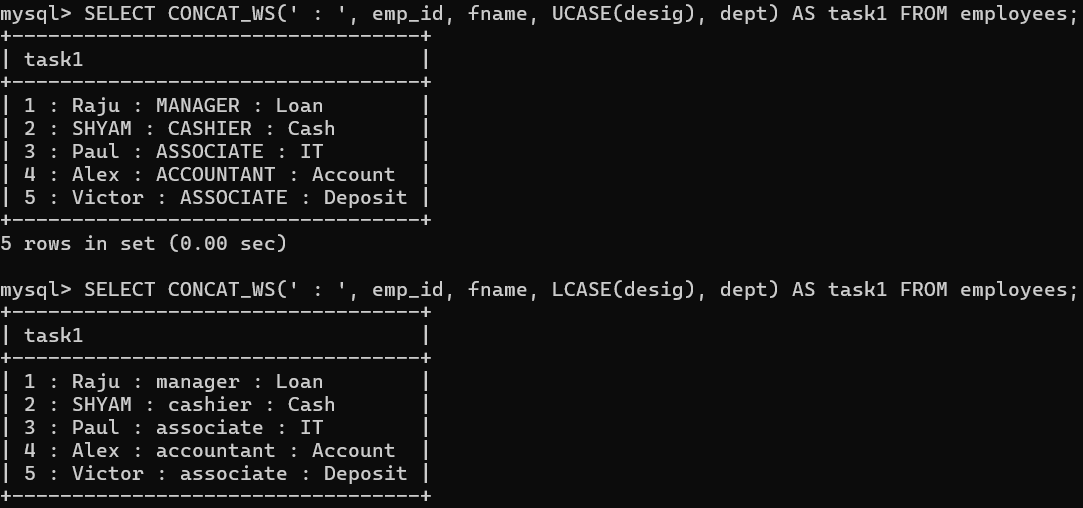
UCASE

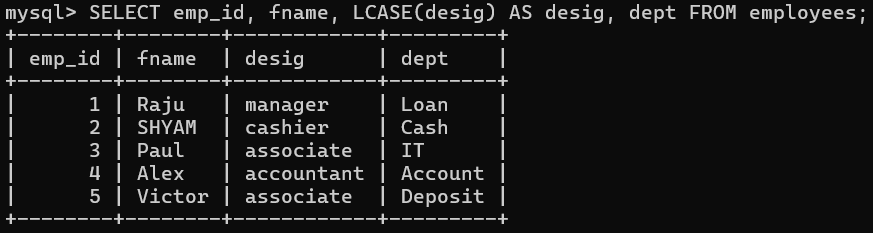
LCASE

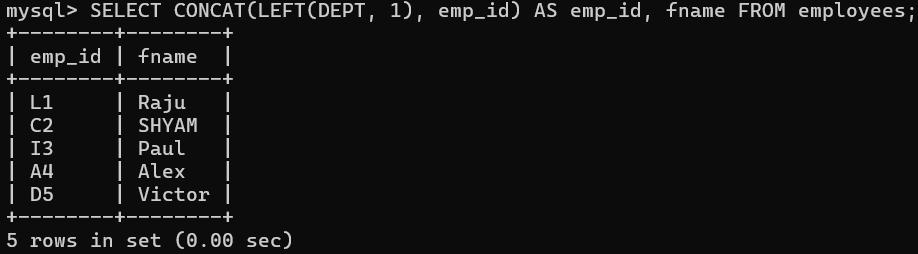








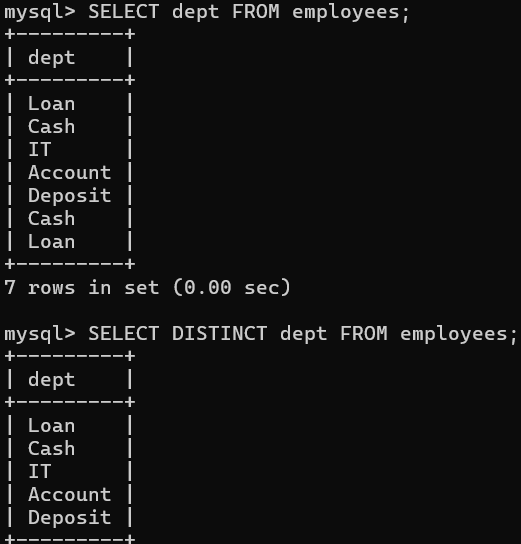




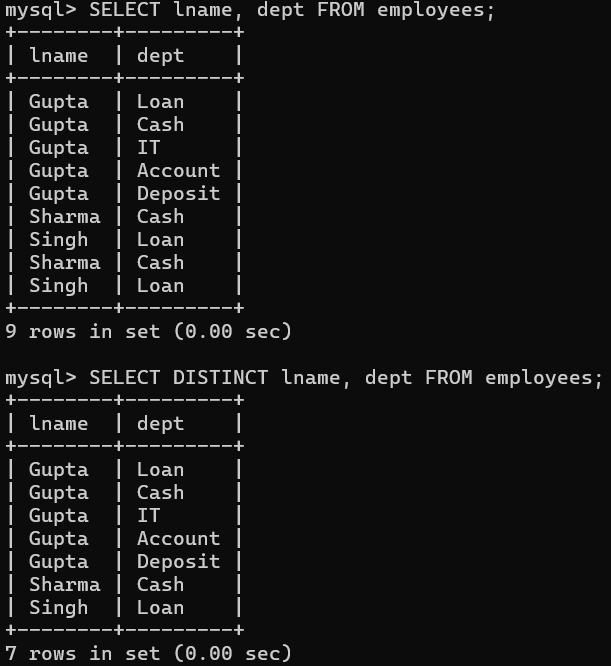
DISTINCT:

Distinct can be use with single column or multiple.

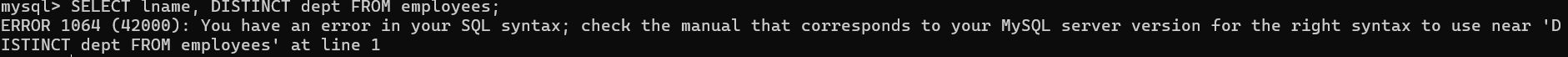
It will return unique value:



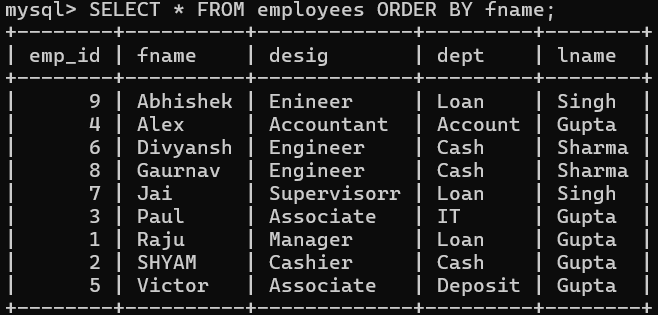
For Mutiple:

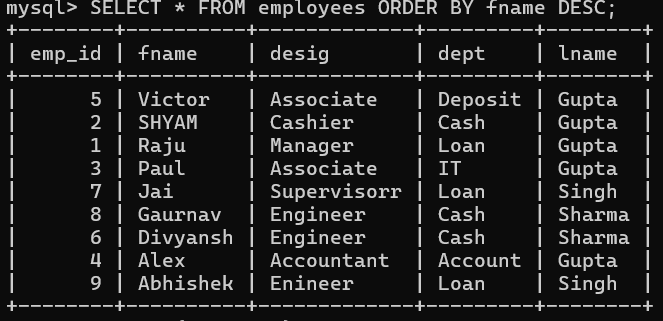


**You cant use distinct between columns**

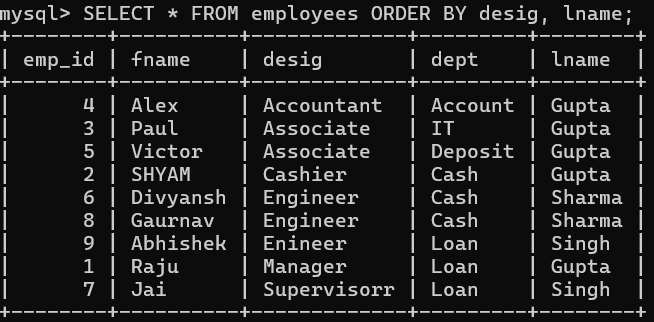


ORDER BY: used for sorting



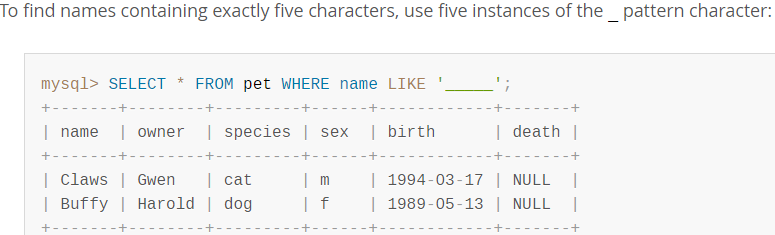
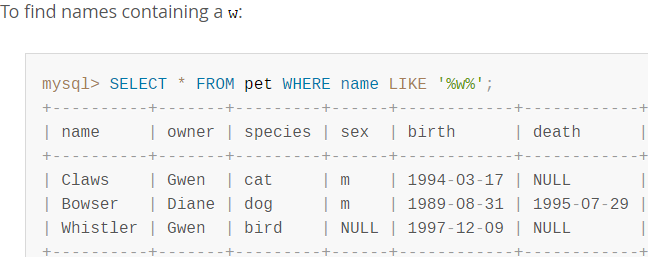
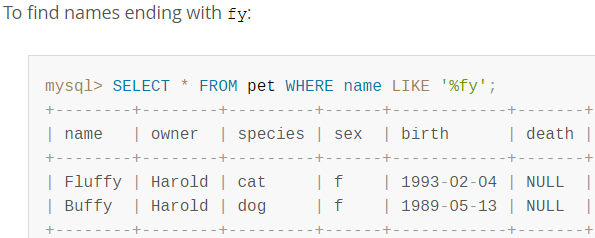
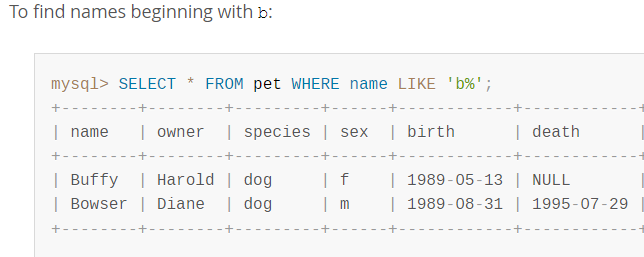


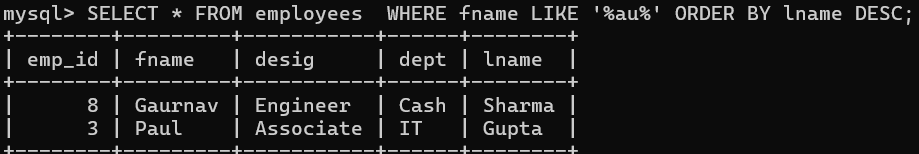
Sorting based on mutiple columns

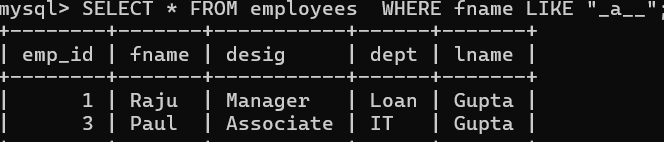


Like:

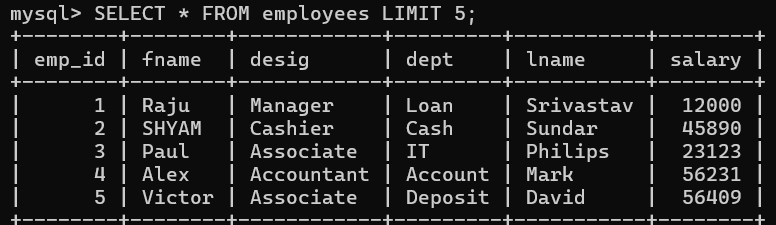
SQL pattern matching enables you to use \_ to match any single character and % to match an arbitrary number of characters (including zero characters). In MySQL, SQL patterns are case-insensitive by default. Some examples are shown here. Do not use = or <> when you use SQL patterns. Use the [LIKE](https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html#operator_like) or [NOT LIKE](https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html#operator_not-like) comparison operators instead.

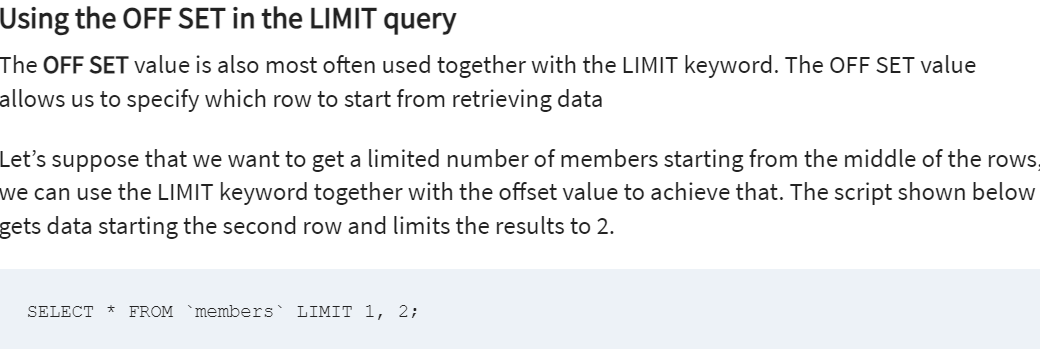


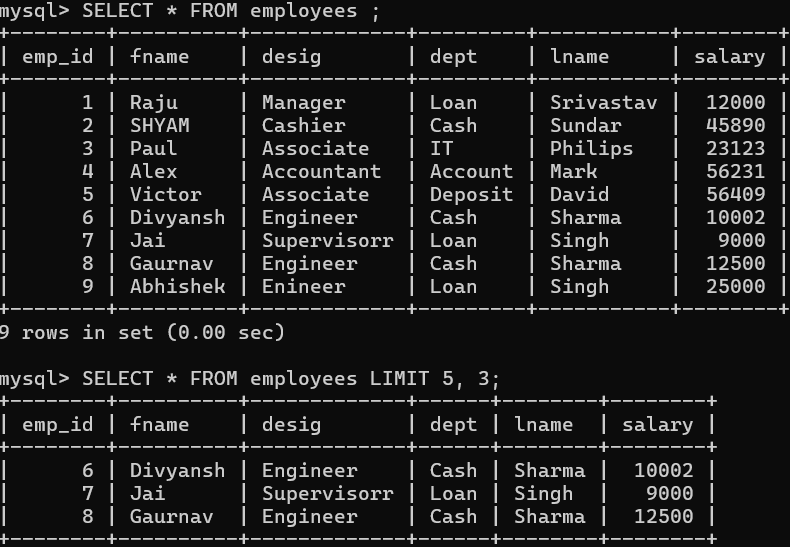




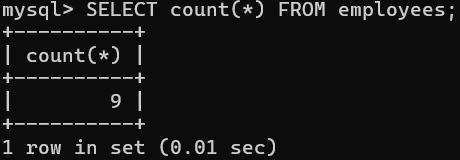
LIMIT

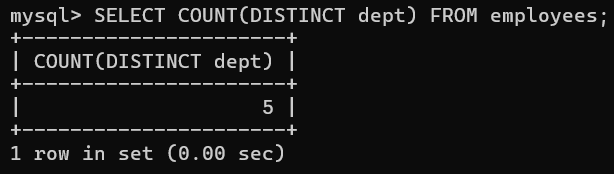






COUNT:



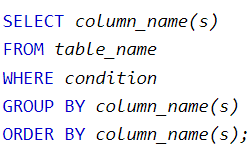


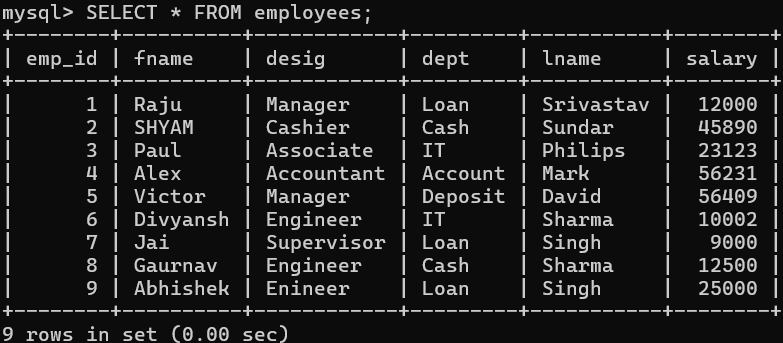
Return count of distinct department

**GROUP BY:**

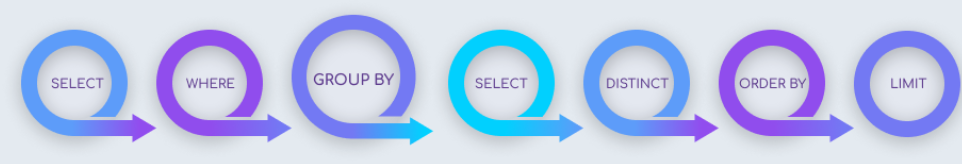
The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

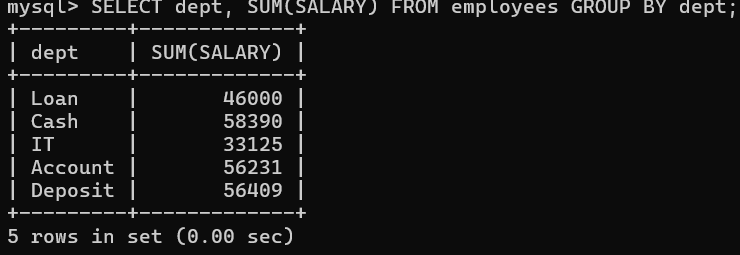
The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.





GROUP BY is one of the most useful MySQL clauses. It groups rows with the same values into summary rows. The clause returns one row for each group. In a query, MySQL GROUP BY is located after the FROM and WHERE clauses, but before the HAVING,  SELECT, DISTINCT,  ORDER BY,  LIMIT clauses.

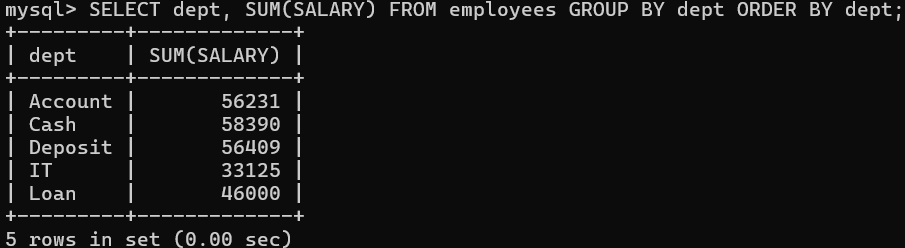




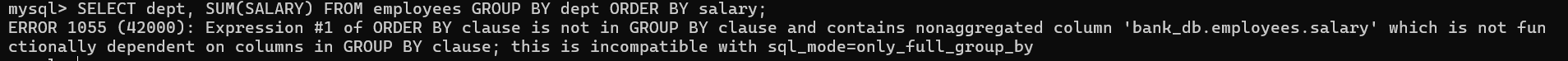
If you use other field except “dept” then you will get error because in grouping it convert all rows with same value together and it dont know which value to use, if we use other field.

**mysql> SELECT lname, SUM(SALARY) FROM employees GROUP BY dept;**

**ERROR 1055 (42000): Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'bank\_db.employees.lname' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql\_mode=only\_full\_group\_by**



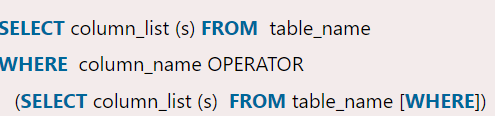
If we use other field in ORDER BY also



**MAX and MIN:**

SUB QUERIES:

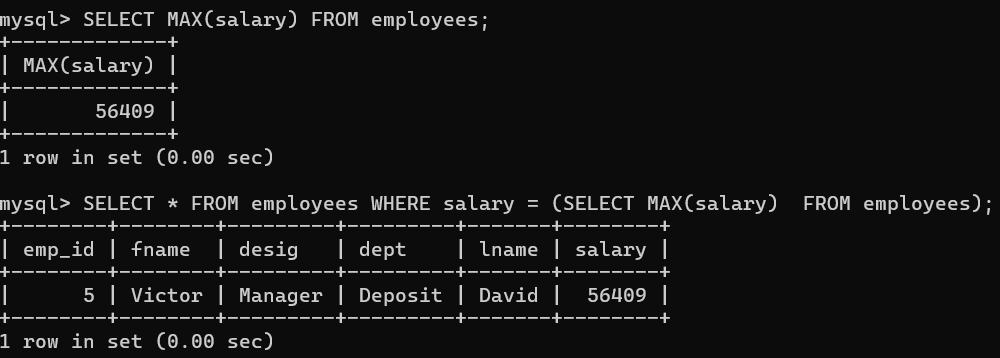
Syntax:

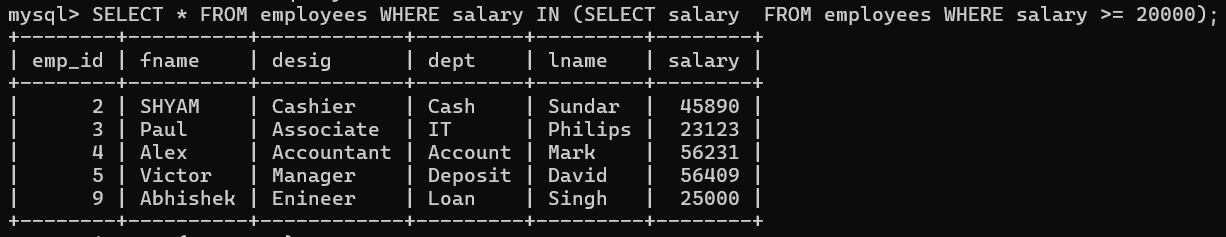


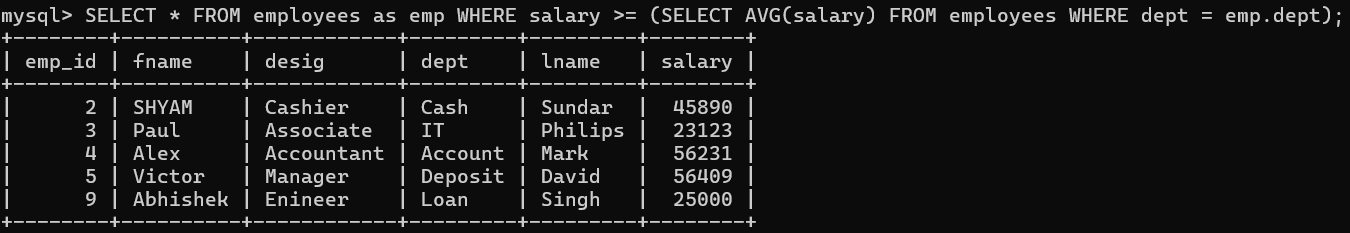
A subquery in MySQL is a query, which is nested into another SQL query and embedded with SELECT, INSERT, UPDATE or DELETE statement along with the various operators. We can also nest the subquery with another subquery. A subquery is known as the **inner query**, and the query that contains subquery is known as the **outer query**. **The inner query executed first gives the result to the outer query, and then the main/outer query will be performed.** [MySQL](https://www.javatpoint.com/mysql-tutorial) allows us to use subquery anywhere, but it must be closed within parenthesis. All subquery forms and operations supported by the SQL standard will be supported in MySQL also.

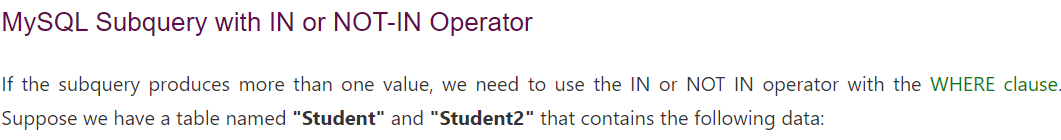
Rules:

* Subqueries should always use in **parentheses.**
* If the main query does not have multiple columns for subquery, then a subquery can have only one column in the SELECT command.
* We can use various comparison operators with the subquery, such as >, <, =, IN, ANY, SOME, and ALL. A multiple-row operator is very useful when the subquery returns more than one row.
* We cannot use the **ORDER BY** clause in a subquery, although it can be used inside the main query.
* If we use a subquery in a **set function**, it cannot be immediately enclosed in a set function.

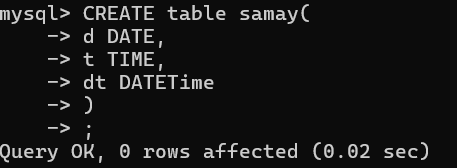


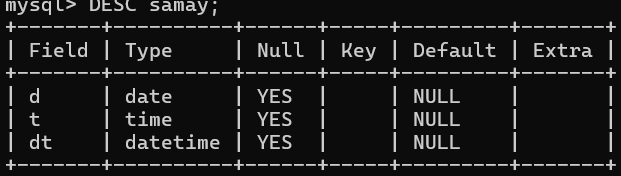


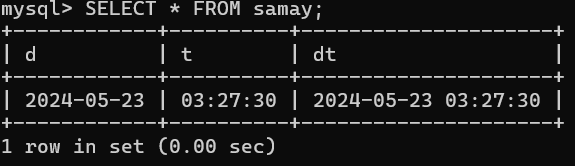




CURDATE(), CURTIME(), NOW():

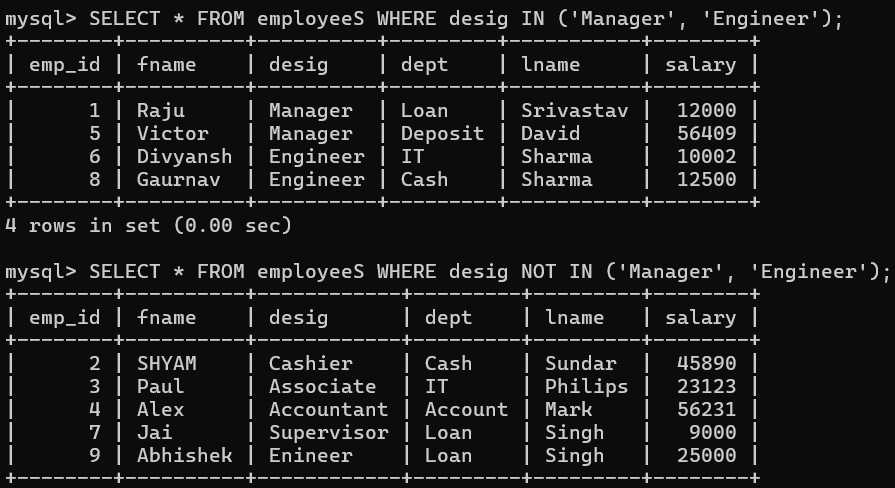




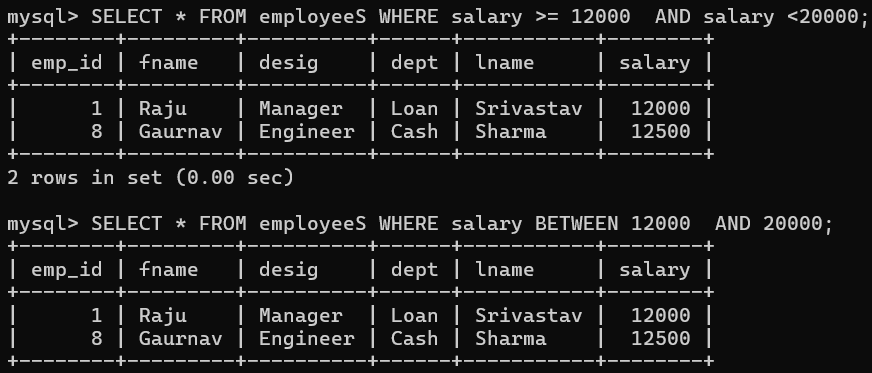


Relational Operator:

IN and NOT IN



BETWEEN and AND



CASE FUNCTION:

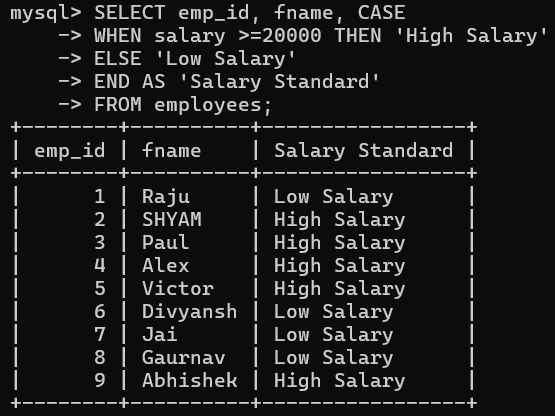
The CASE statement goes through conditions and return a value when the first condition is met (like an IF-THEN-ELSE statement). So, once a condition is true, it will stop reading and return the result.

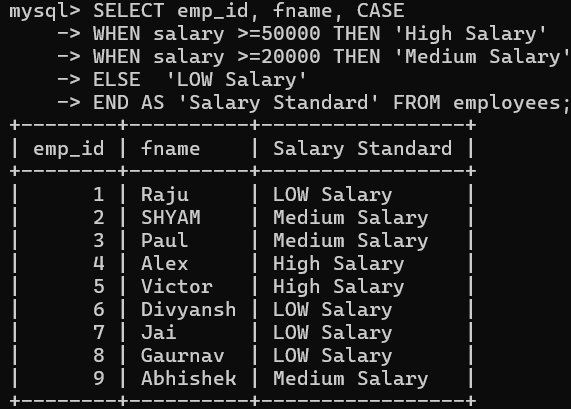
If no conditions are true, it will return the value in the ELSE clause.

If there is no ELSE part and no conditions are true, it returns NULL.

## Syntax

CASE  
   WHEN *condition1* THEN *result1*   WHEN *condition2* THEN *result2*   WHEN *conditionN* THEN *resultN*   ELSE *result*END;





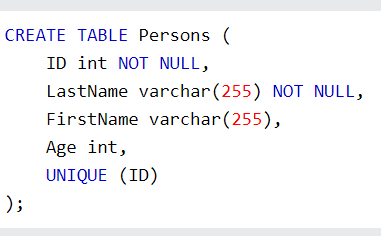
UNIQUE:

The UNIQUE constraint ensures that all values in a column are different.

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

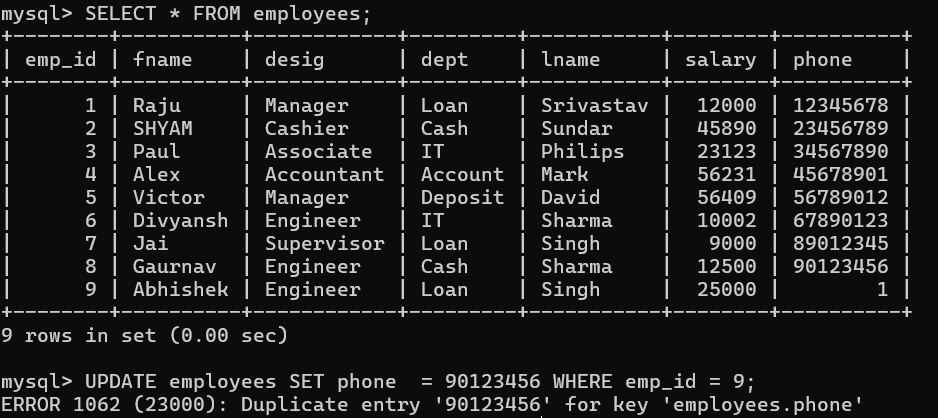
A PRIMARY KEY constraint automatically has a UNIQUE constraint.

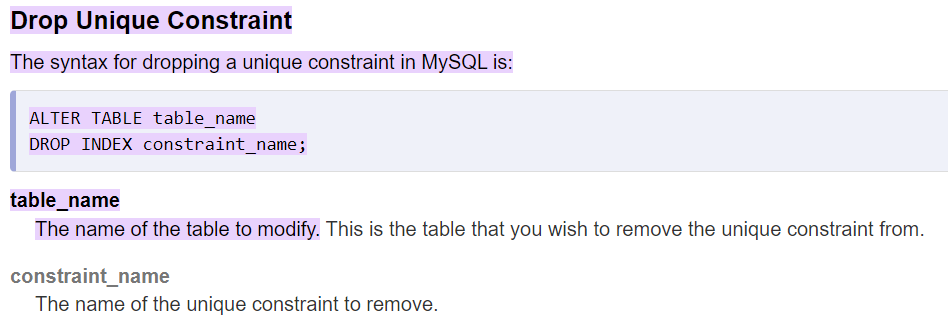
However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

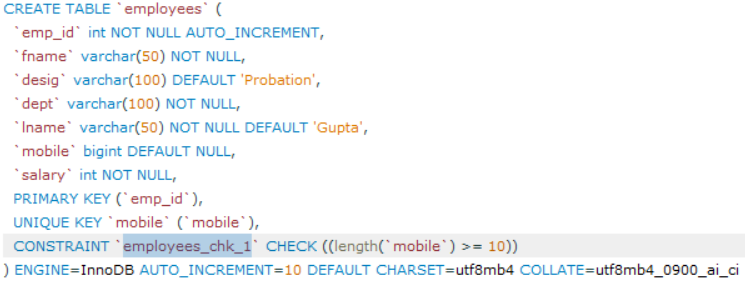


Why we use UNIQUE if we have PRIMARY?

It is because a table can have only one PRIMARY KEY and if we want to make the value of column unique then we have to use UNIQUE constraint.



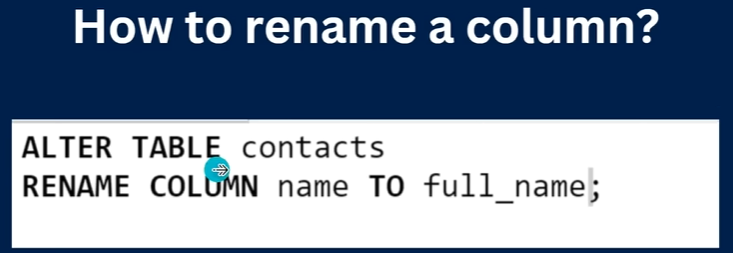


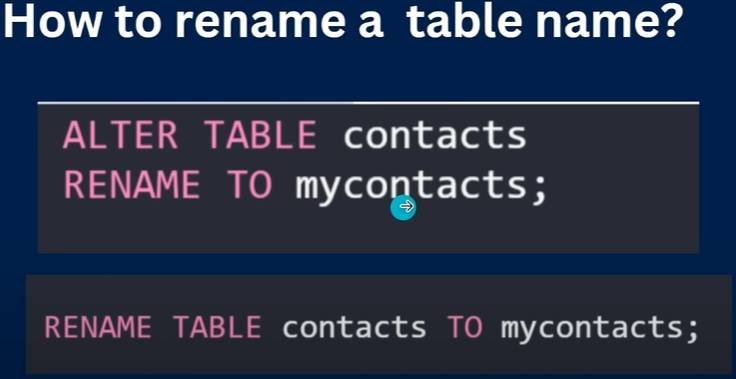


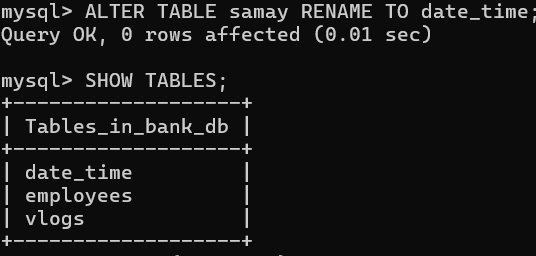
Here you can change constraint name “employees\_chk\_1” to some meaningful name.

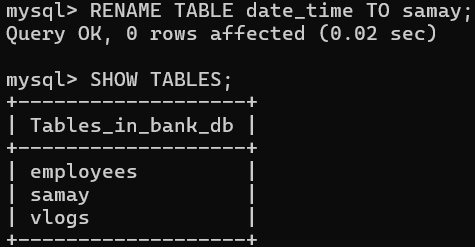


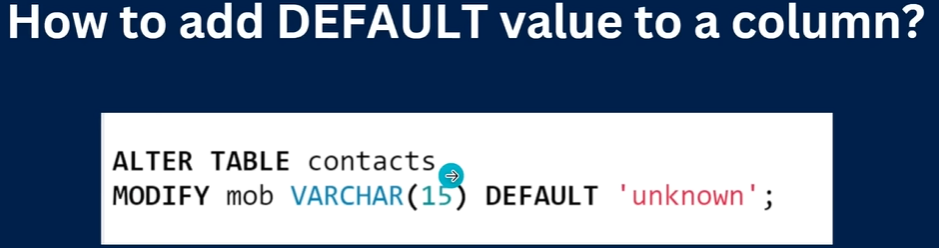




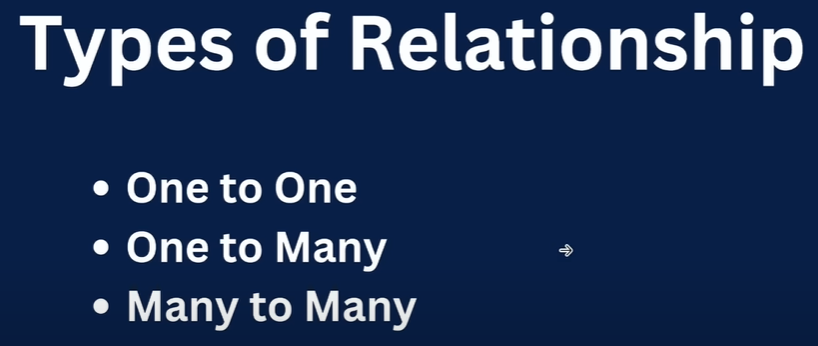


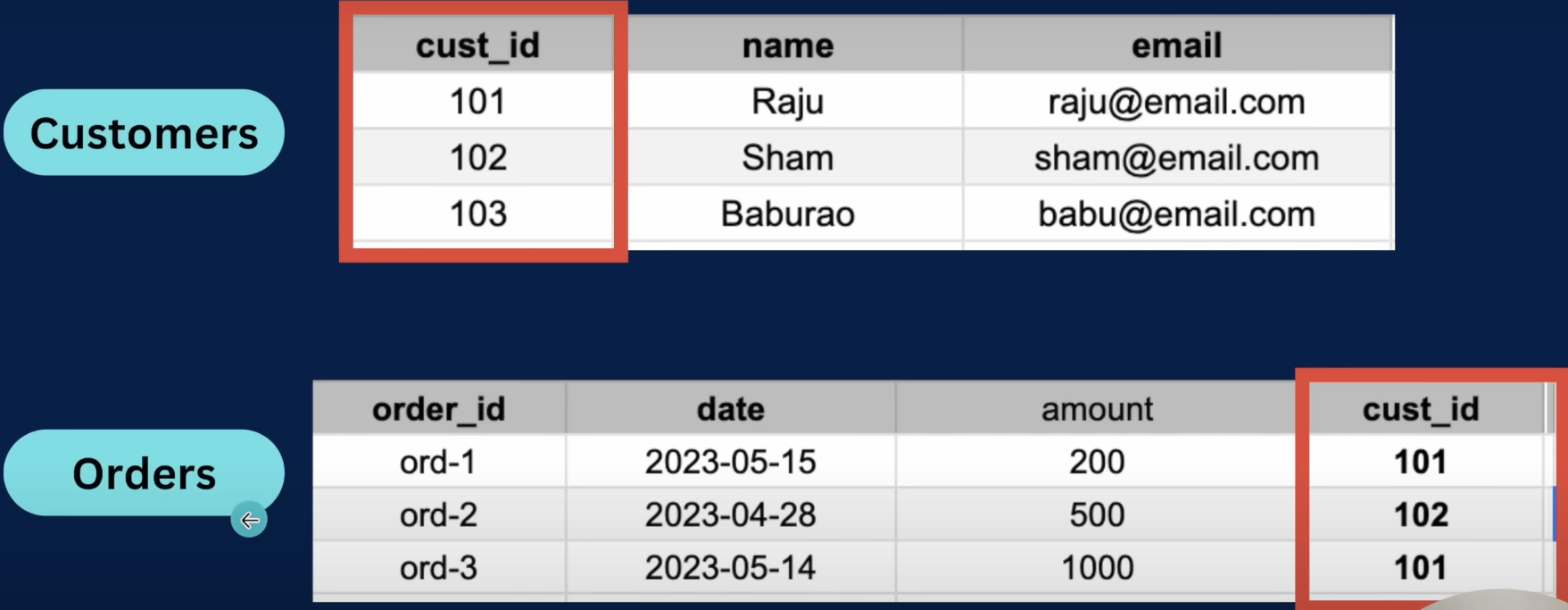






Relationship in multiple tables





In above tables we have a relation based on customer id ’cust\_id’.

OneToMany relation

One -> customer

Many -> Orders

A customer can order multiple orders.

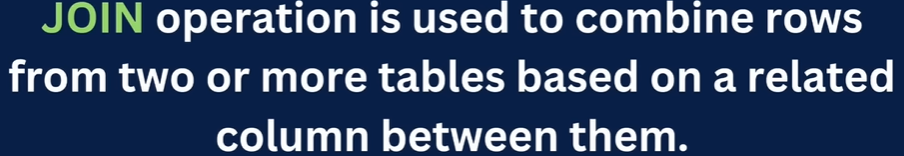
NOTE:

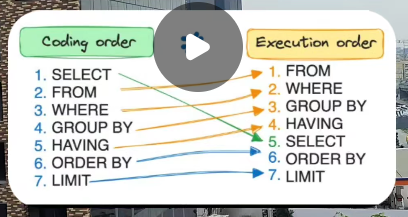
It is mandatory to give string size in VARCHAR(SIZE) else throw error.

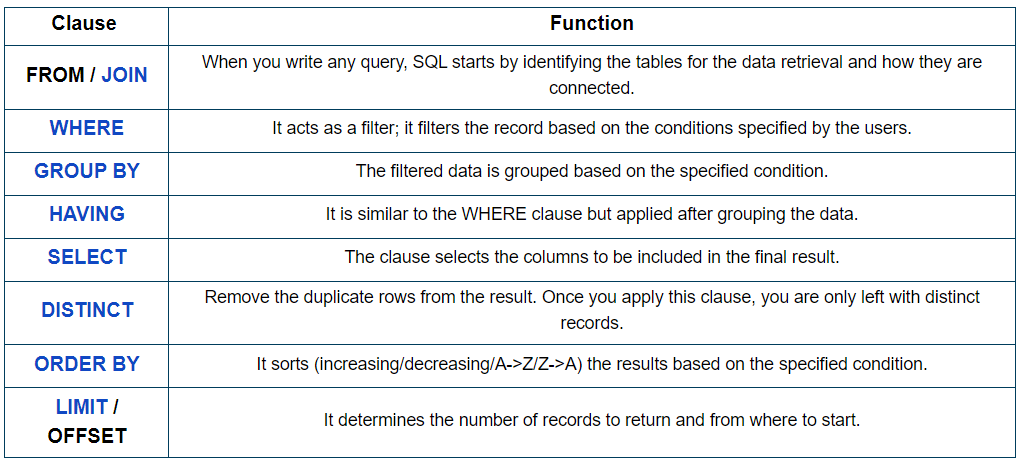
CONSTRAINT phone\_number\_should\_be\_10\_digits CHECK((LENGTH('mobile') = 10)));

Here **phone\_number\_should\_be\_10\_digits** should not be in single quote, double quote will throw error.

JOIN:







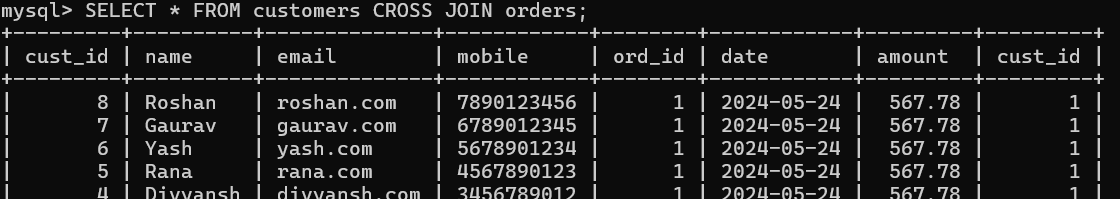
<https://matam-kirankumar.medium.com/sql-query-order-of-execution-37001da1462>



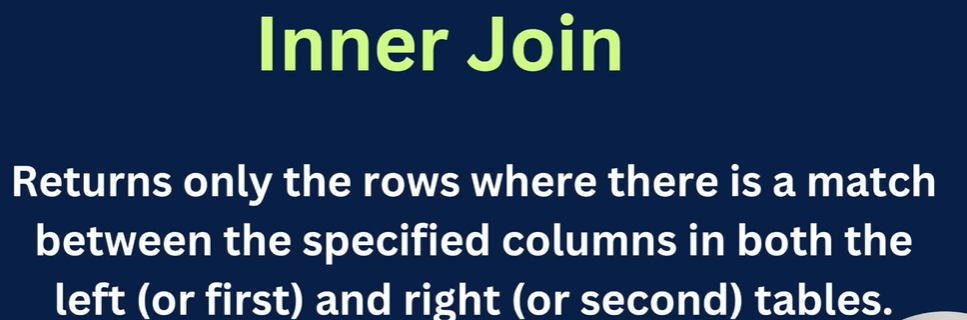
**SELECT -> DISTINCT -> FROM -> JOIN -> ON -> WHERE -> GROUP BY -> HAVING -> ORDER BY -> LIMIT -> OFFSET**

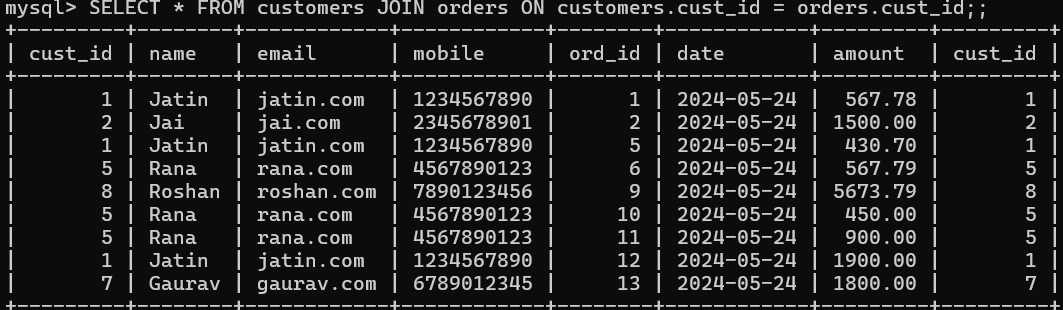
CROSS JOIN:





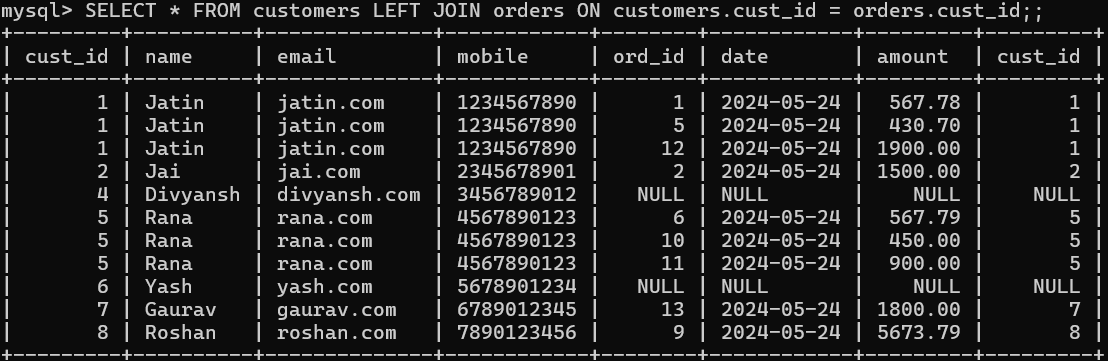
INNER JOIN:



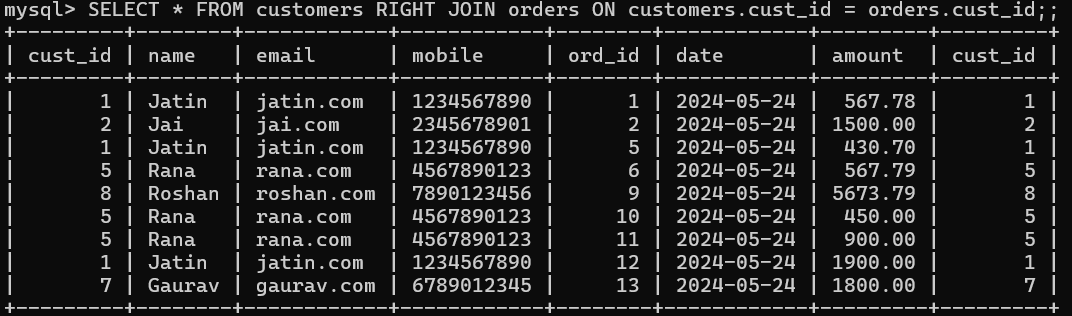


LEFT JOIN:

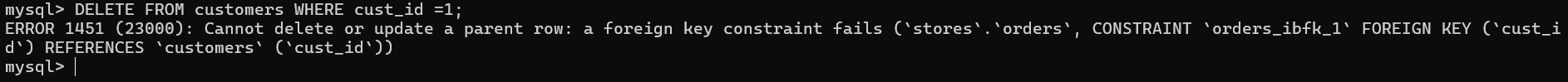




RIGHT JOIN:

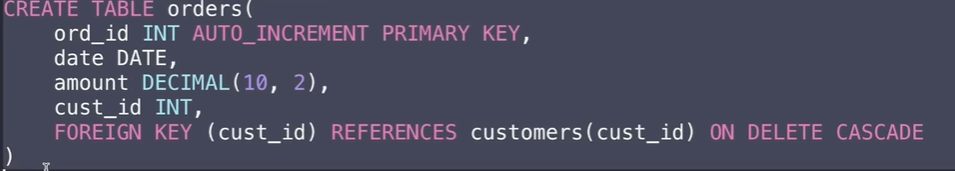


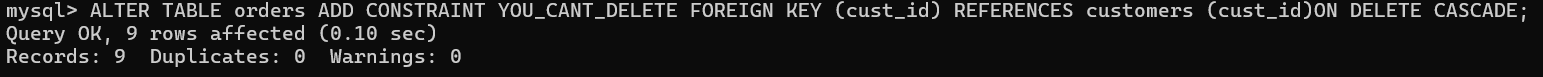
ON DELETE CASCADE:



You cant delete parent row because it is used in other table as a foreign key.

To delete this you have to use “ON DELETE CASCADE” at the time of table creation.

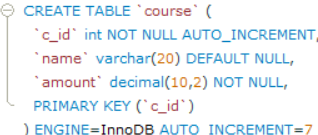
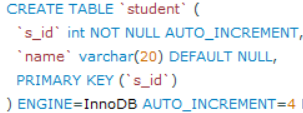




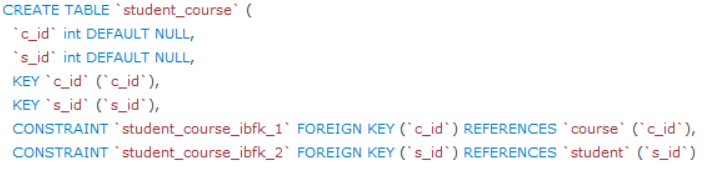


MANYTOMANY:

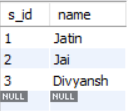
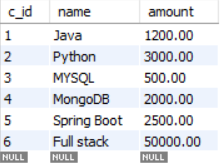
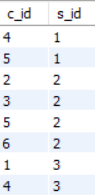
course table: student table:

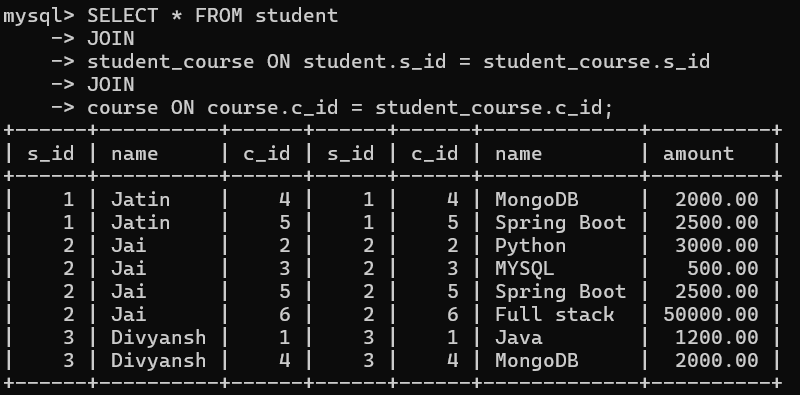
 

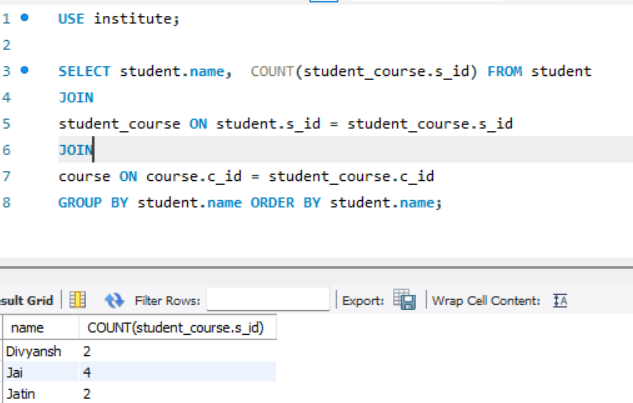
student\_course table:

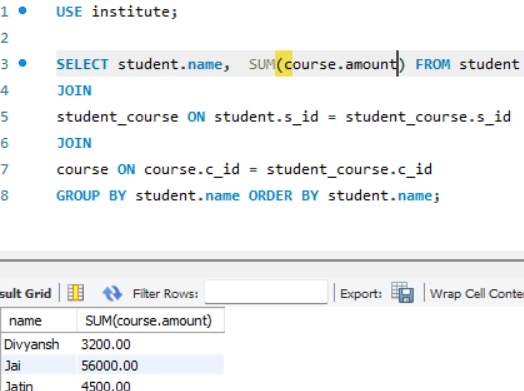


student course student\_course

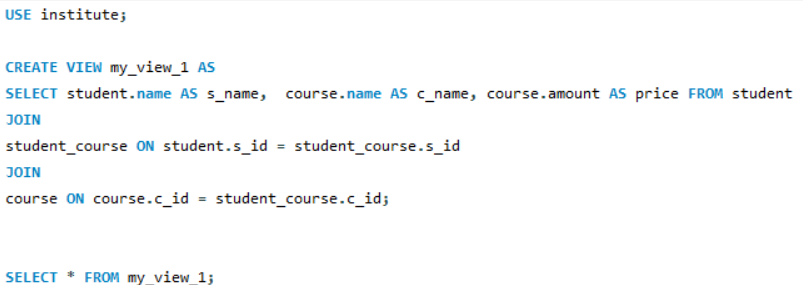






First insert data in student, then course and then student\_course.

In the above query you namt return many lines of code and some are same in all queries, to refrain from these you can use **VIEW** which will create virtual view.



Here I created “**my\_view\_1**” for below query and then i am fetching data from view.

“**my\_view\_1**” will be created.