MySQL Notes

Database: It is used to store the data

DBMS: Stands for Database Management System, it is a software that can access the database, like file system, relational database, queue, and so on

RDBMS: It is a kind of DBMS that maintains the data in a table format.

There are many RDBMS they are:

* MySQL
* Oracle
* Derby
* Sybase
* MSSQL
* IBMDB2

MySQL: It is a open source database it is from Oracle, it is a RDBMS that can store the data in a table format.

RDMBS understands a language called SQL to communicate with the database

SQL: It stands for Structured Query Language, which is a language the relational database understands, this gives queries to the relational which are nothing but commands to perform various operations like CRUD operations i.e., Create, Retrieve, Update & Delete

Since SQL can perform various CRUD operations, it has divided into 5 sublanguages the are:-

1. DDL - Data Definition Language(create, alter, truncate, drop) - Deals with the structure of the database table like creating, altering, dropping, truncating records
2. DML - Data Manipulation Language(insert, update, delete) - Deals with data/record manipulation like inserting, updating & deleting
3. DRL/DQL - Data Retrieval/Query Language(select) - Deals with data/record retrieving
4. TCL - Transaction Control Language (rollback, commit) - Deals with saving or undoing DML operations
5. DCL - Data Control Language (grant, revoke) - Deals with granting/revoking permissions to the users

SQL provides lot of features while maintaining the data

* datatypes
* constraints
* inbuilt functions
* keywords to perform various operations

Datatypes:

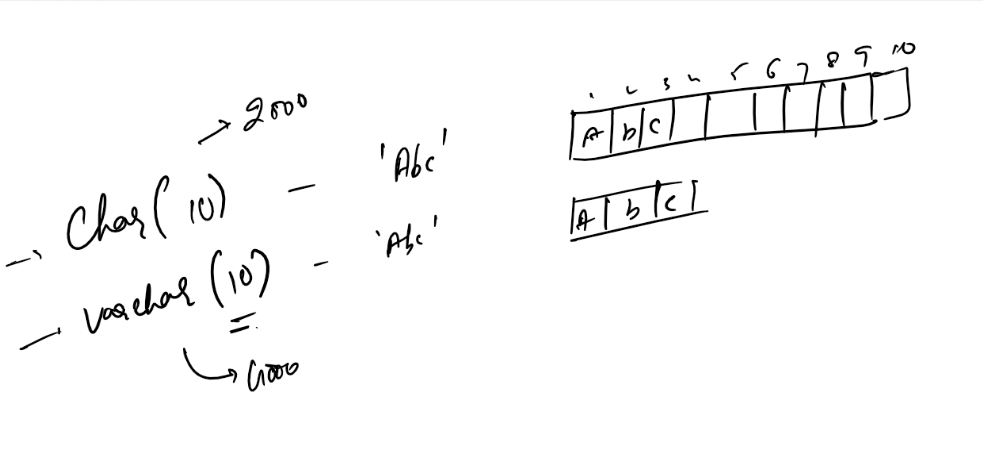
MySQL has its own datatypes

* int
* bigint
* varchar
* double
* date
* datetime
* char
* boolean

int vs bigint: They both accept integer values, but bigint can represent bigger values than int in terms range

char vs varchar: char & varchar both are used to store characters & text, but char is fixed, varchar is dynamic i.e., char(5) - allocates memory for 5 characters, if you store a single character then it still takes memory of 5 characters in char, however in varchar(5), it can still take maximum 5 characters, if you store a single character it allocates memory only for 1 character.

char supports maximum 2000 characters & varchar supports 4000 characters



date vs datetime: date can only store date in yyyy-MM-dd format i.e., 2022-03-15, date time stores both date & time in yyyy-MM-dd hh:mm:ss format i.e., 2022-03-15 12:03:55

Constraints: These are the rules that can be applied on a table or a column like value must be unique, value must be within some range, value must depend on another value, value is required an so on

There are 5 constraints in MySQL which are keywords as well

* primary key
* unique
* check
* not null
* foreign key

Primary key

* It is given to a column to uniquely identify the entire record
* it can never be null, value is mandatory
* It is mainly used to give higher priority to access the record, while searching or updating records
* Students rollno, Customer’s account number, employees id all these can be considered as a primary key
* Duplicates not allowed in primary key, if you do that database raises error

Unique:

* It is given to a column to have unique values
* It can have null, i.e., value is not mandatory but if you are giving value it must be unique
* It is going to have a secondary priority compare to primary key while searching or updating records
* phone, aadhar numbers, pan, all these can be considered as a secondary key to recognize the record

Check:

* It is mainly used to check the values of the column adhere to the rules
* Suppose you want values to be within some range
* age must be > 18 < 60, gender must be either male or female and so on.
* Check constraint works only if you have MySQL 8.0.16 or higher

Not null:

* When you want a column to mandatorily have a value then you can use not null

Foreign Key:

* This is a column whose value depends on another table primary key column value
* It allows to have records in different tables and related them with a single column foreign key

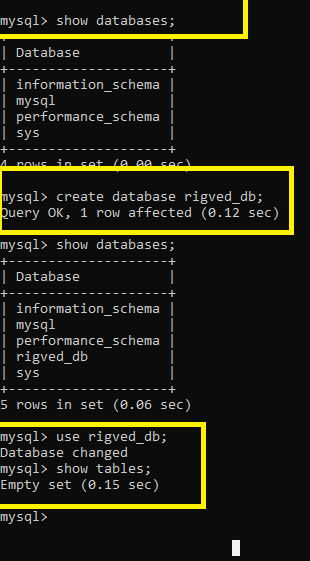
SQL commands

>> create database database\_name;

>> use database\_name;

>> show databases;

>> show tables;

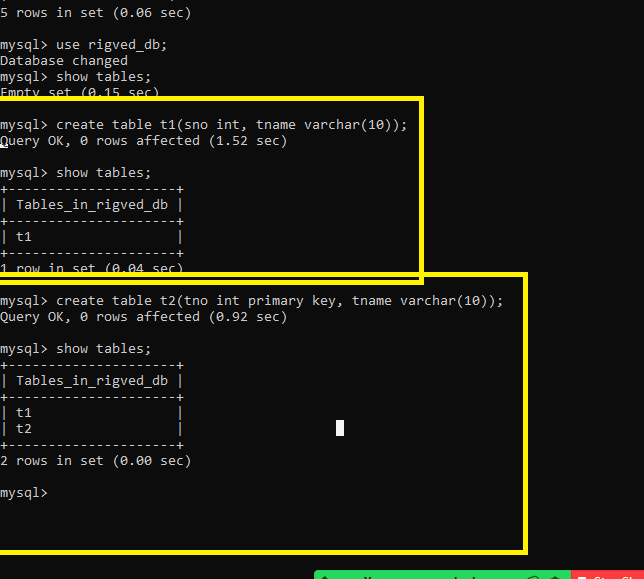


Create table:

Syntax: create table table\_name(column\_name *type(size)* constraint, column\_name *type(size)* constraint,…);

Note: size may not be necessary in some types like double, int, date and so on

Note: constraint is optional, but its recommended to create primary key as it recognizes each record uniquely



Insert records

To store the records in a table you can use insert command

syntax: insert into table\_name values(value, value,…);

syntax: insert into table\_name(column, column, …) values(value, value,…);

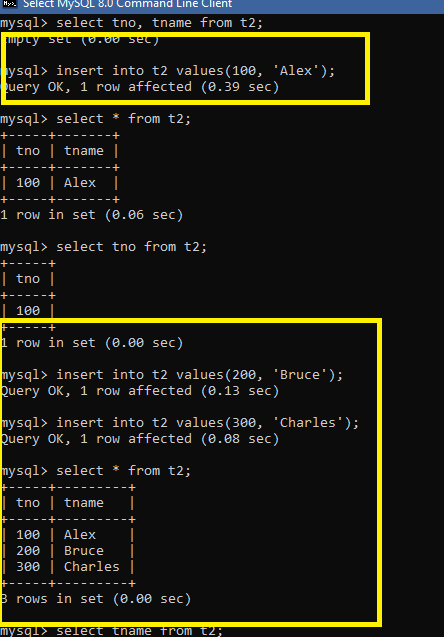
Fetch records

To fetch the records in a table you can use select command

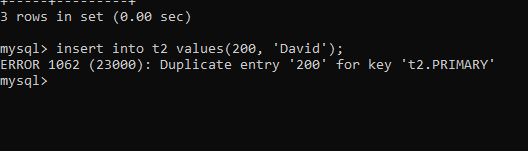
syntax: select column, column,…. from table\_name;

syntax: select \* from table\_name;

Insert & Select command



What happens if the tno is duplicate



Note:

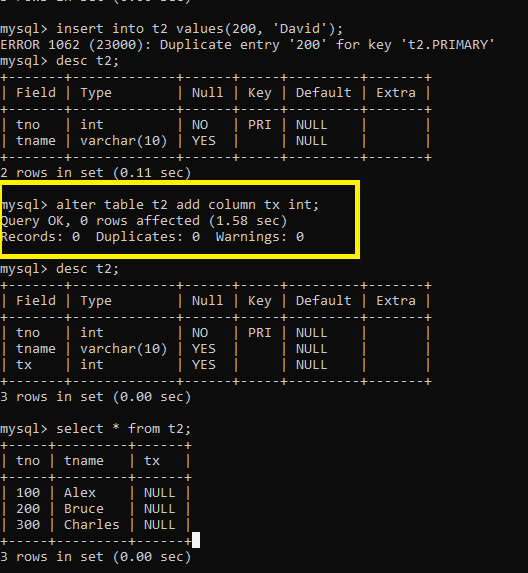
* Create is a DDL
* Insert is a DML
* Select is a DRL/DQL

ALTER command

It is used to modify the existing structure of the table, like adding a column or removing a column or changing name of the column, add/remove constraints

Syntax to add a new column

alter table table\_name add column column\_name datatype;



Activity:

1. Try above activities
2. Create a table employee which will have following column names and the constraint

employee\_id - primary key

employee\_name - not null

age - check constraint which must accept value between 18 and 60

email - unique constraint

1. Store the records in the above employee table & try to store age 70 and age 17 and see what happens
2. Store the few more records in the above employee table & try to store same email to multiple records and observe what happens
3. Alter the above employee table add two new columns pan and aadhar
4. Alter the above employee table and add unique constraints to pan and aadhar
5. Insert few records in employee table and observe the result, try to pass duplicate aadhar or pan and observe the result.
6. Alter the above employee table and remove the pan and aadhar columns
7. Observe the result after removing the columns
8. Using the DROP delete the employee table.

Truncate:

It deletes all the records from a table but retains the table structure, truncate command can’t be rolled back, i.e., it is permanent.

>> truncate table table\_name;

Drop:

It deletes the table permanently

>> drop table table\_name;

Delete:

It deletes the records from the table, it can have conditions while deleting so that only the records that matches to the condition would be deleted, delete command can be rolled back, i.e., you can get the data deleted back

>> delete from table\_name <<condition>>

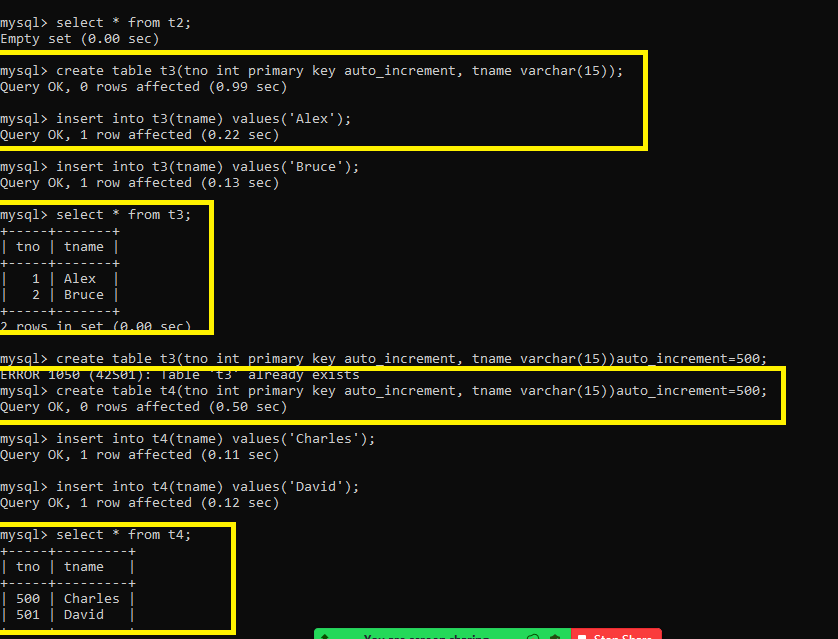
|  |  |  |
| --- | --- | --- |
| Truncate | Drop | Delete |
| It permanently deletes the record | It deletes the table permanently | It deletes the record |
| You can’t write conditions | You can’t write conditions | You can write conditions to delete |
| Can’t be rolled back | Can’t be rolled back | Can be rolled back |

Auto\_Increment:

It is used to automatically increment the values, it is used for a primary key, it is to auto generate the numbers, like employee id, customer id, student roll no, transaction id and so on.

syntax: create table table\_name(column primary key auto\_increment, …);

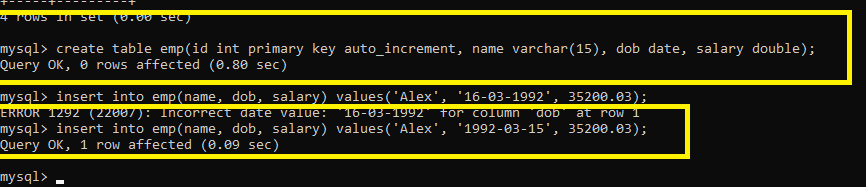
syntax: create table table\_name(column primary key,…) auto\_increment=value;

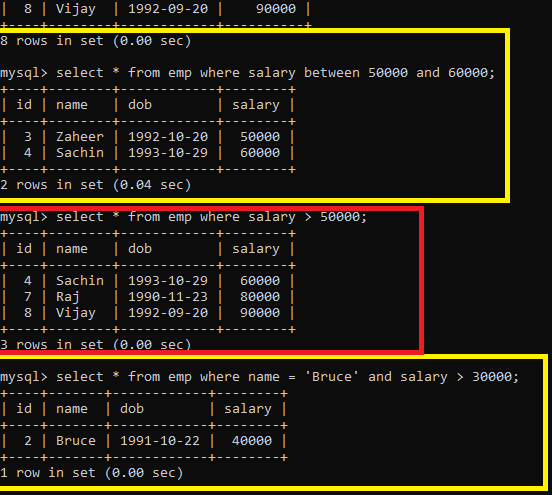


DRL/DQL:

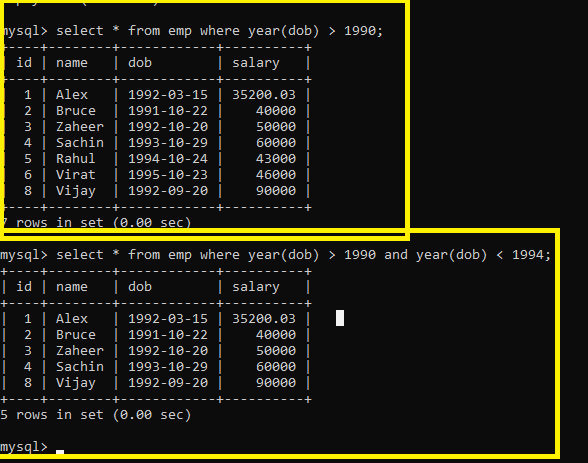
It is used to fetch the records by applying various conditions and keywords, SQL uses lot of operators for conditions like

=, <=, >=, !=, where, in, and, or, between, join, left join, right join, group by , order by, limit





Selecting the records using year() function



SQL Inbuilt functions

There are two types of functions

1. Single row functions
2. Aggregate functions

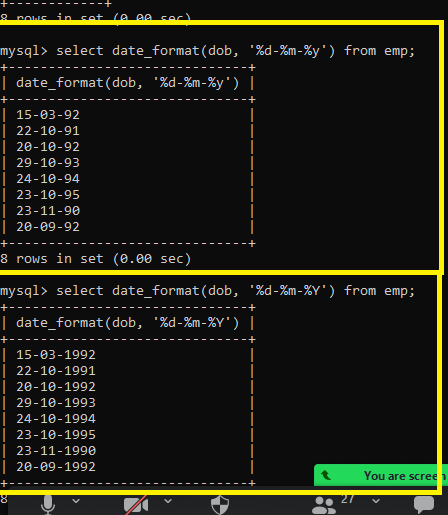
Single row functions: The function is applied on each row and returns the result for each row

ex: year(), date\_format, upper, lower, concat, time\_format, substring, replace

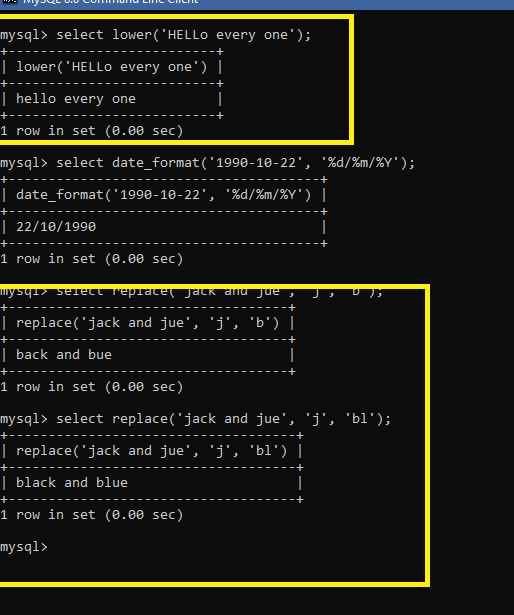
Aggregate functions: The function is applied on each row and returns the single result by applying it in all the rows

ex: avg, sum, count, min, max,

Single row function on emp

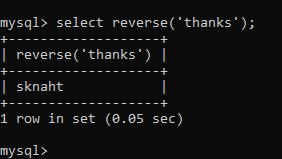


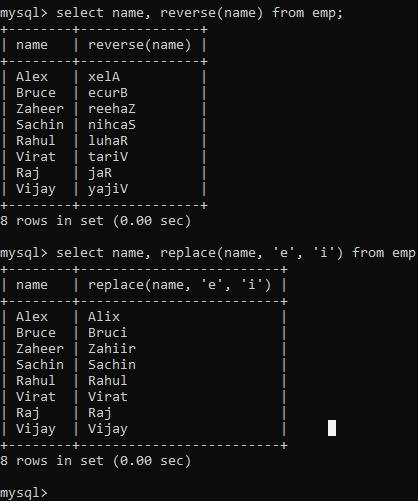
You can use single row function on a data directly without using a table



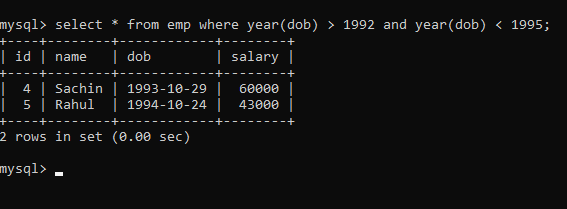
replace: it replaces the string with other characters, 1st argument is the string, 2nd is the matching string, 3rd is the string that replaces the matching string

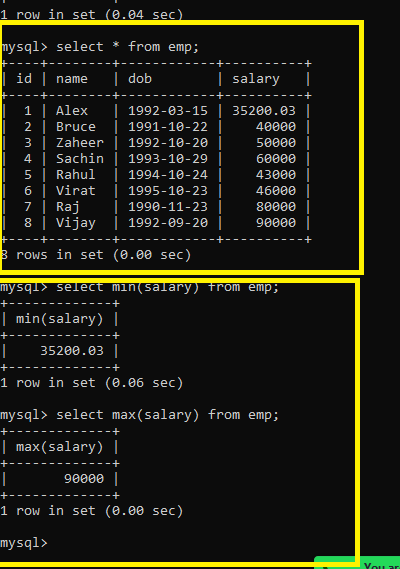
reverse: This will reverse the string



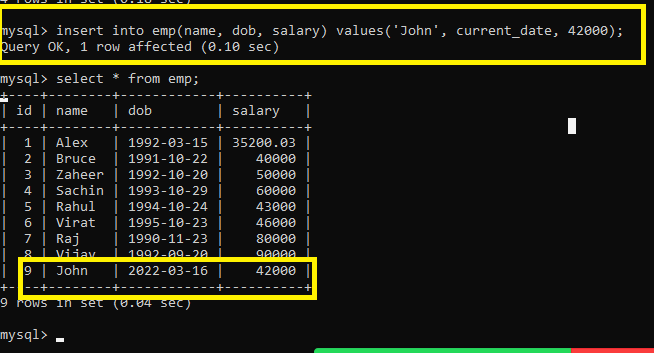


year(): to apply condition on date year

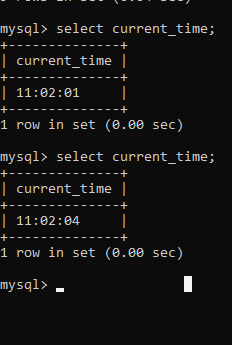
Aggregate functions



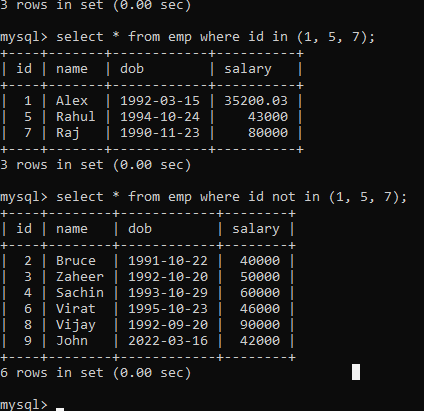
current\_date while storing



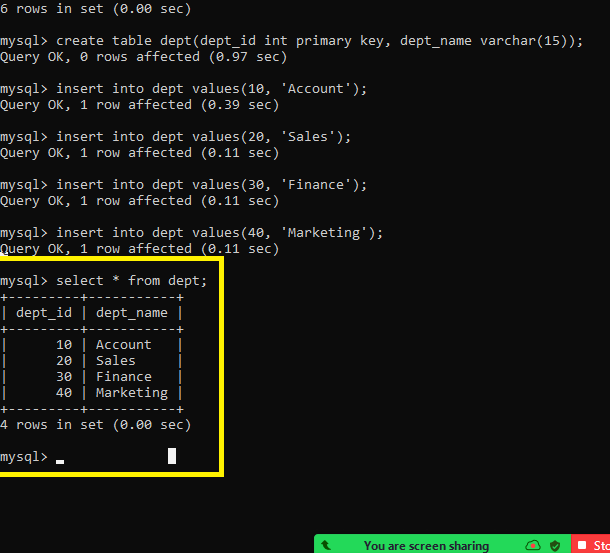
You can use current\_time to store the data



IN operator: It is used to write conditions on multiple values

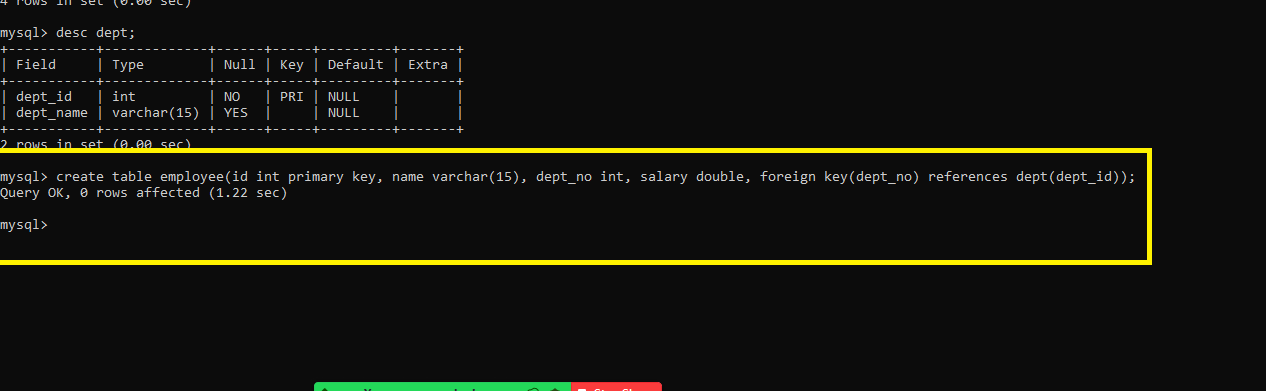


Creating a dept table to link with another table employee

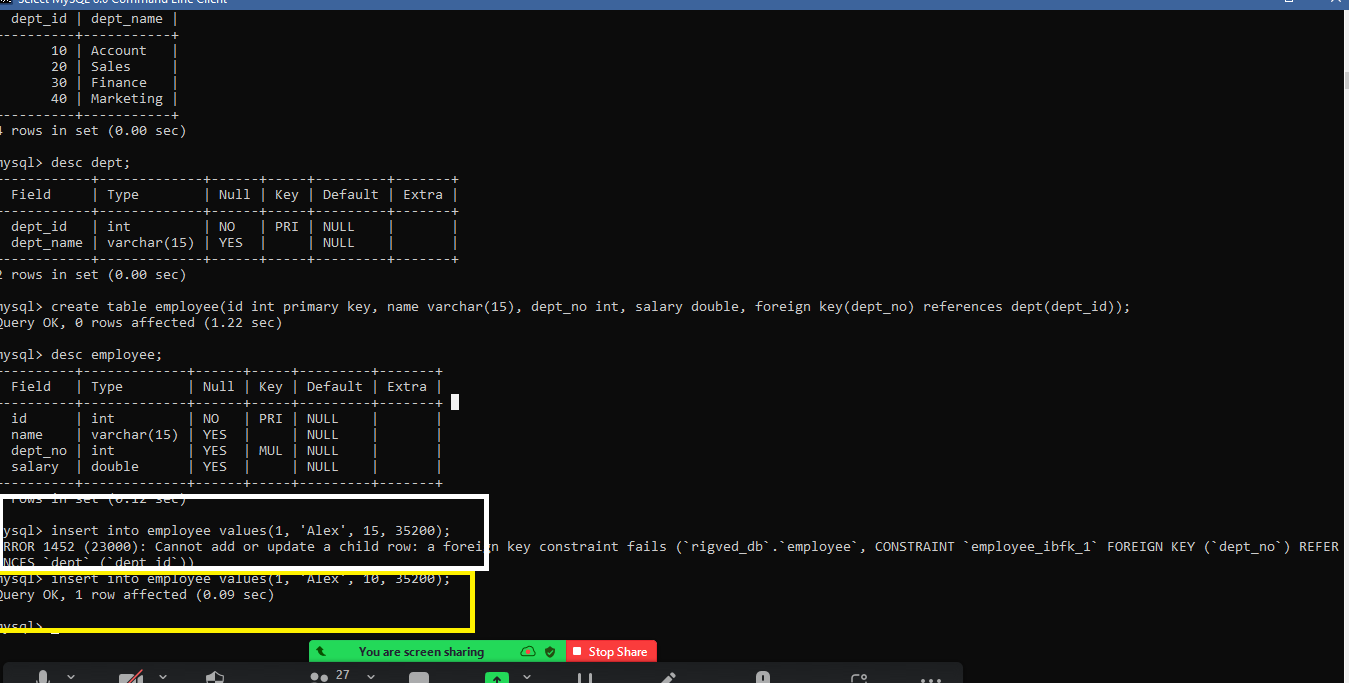


Foreign key: It is a column that references another table primary key column, so it always depends on another table data

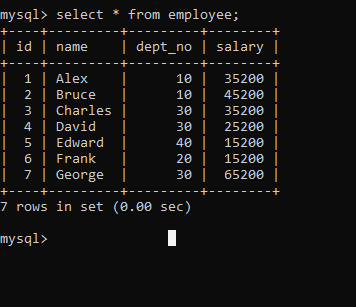
creating an employee table with foreign key



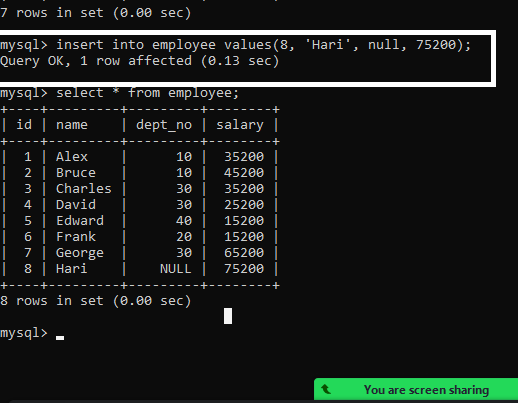
Now the employee table references to the dept table, the employee dept\_no is a foreign key whose value must be present in the dept table or it could be null.



Since 15 is not present in the dept table it throws error while storing employee information, however 10 is there in dept hence it doesn’t throw error.

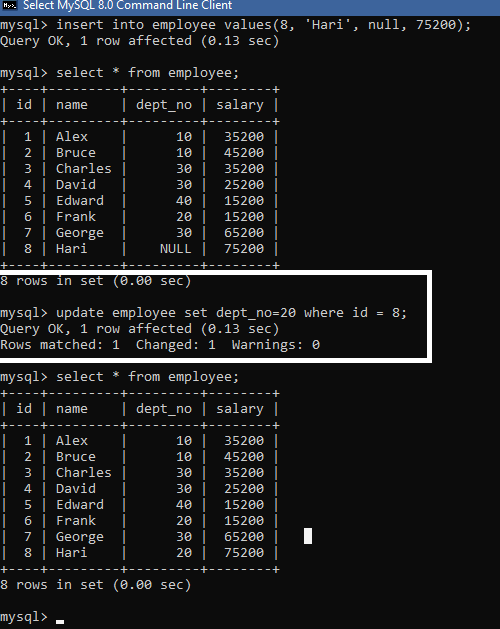


You can add null to foreign key column



You can update the department using update query

update table\_name set column=value where <<condition>>

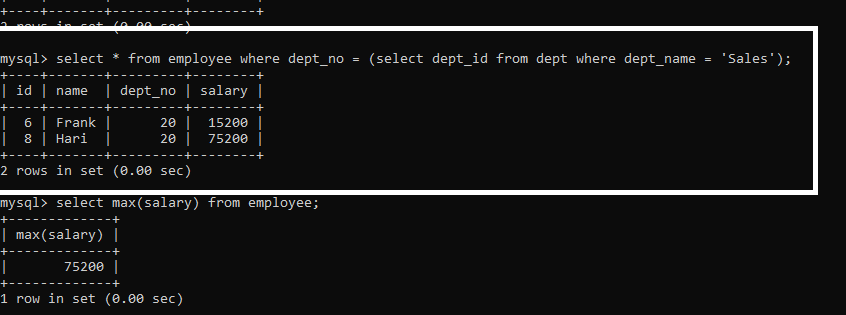


Subquery/Nested queries/inner queries:

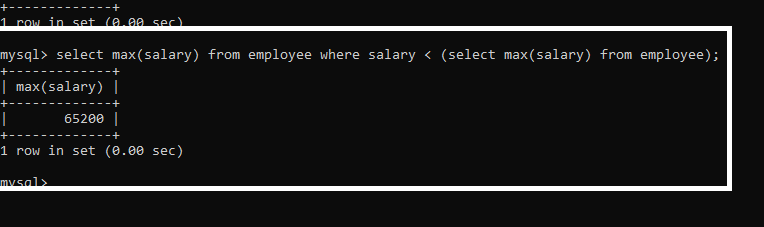
These are the queries which is used to apply condition on an outer query

syntax: outer query condition (inner query)

Selecting employees belonging to some department

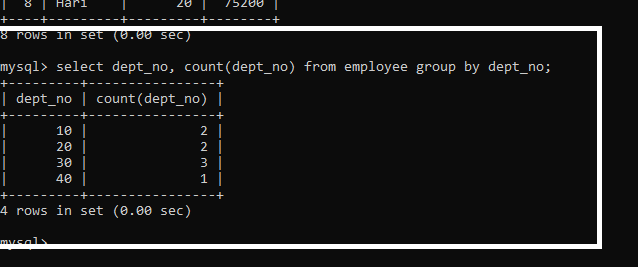


Finding the second maximum salary



Group by

It is used when you want to group results with aggregate functions



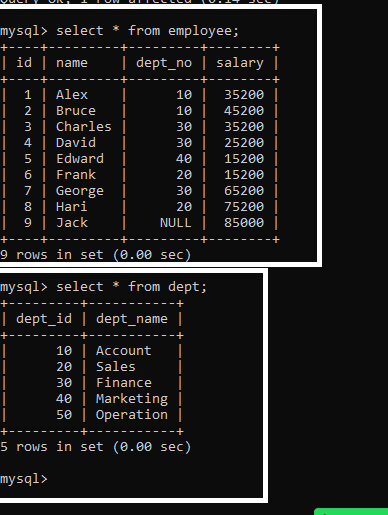
SQL JOIN

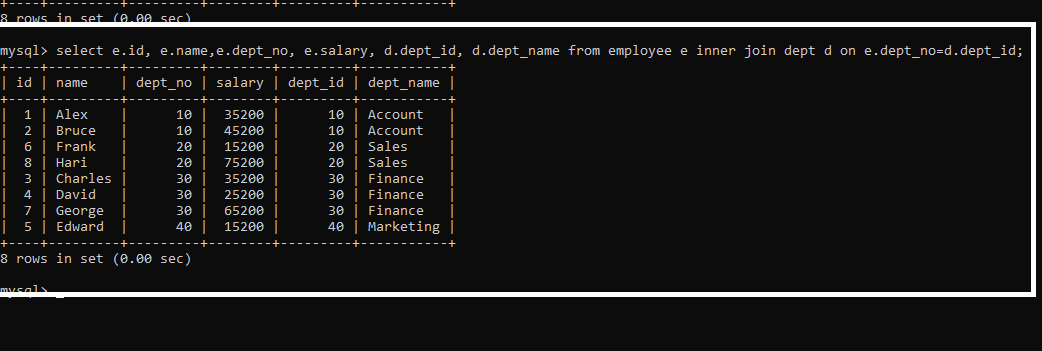
It is used to join 2 or more tables to obtain a result, ex: you want to join employee & department & show both the tables result as a single result

There are 4 types of joins

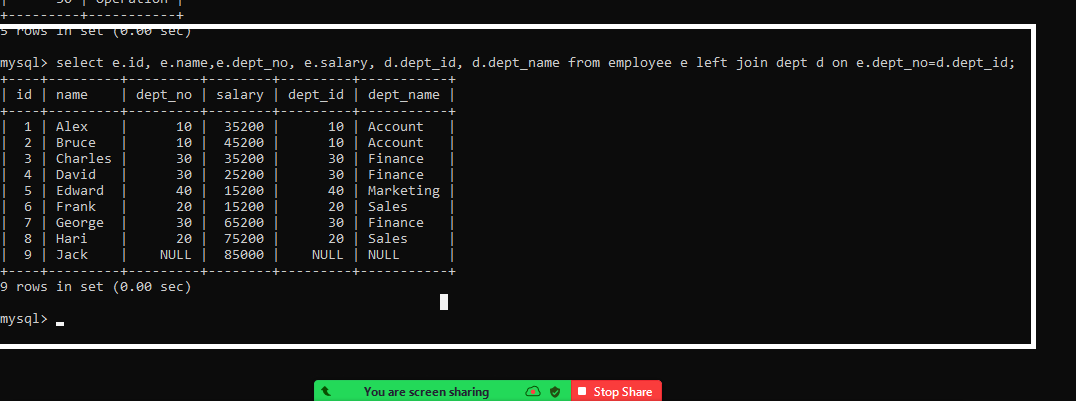
1. Inner JOIN
2. Left JOIN
3. Right JOIN
4. Full JOIN

Current employee & department looks like below

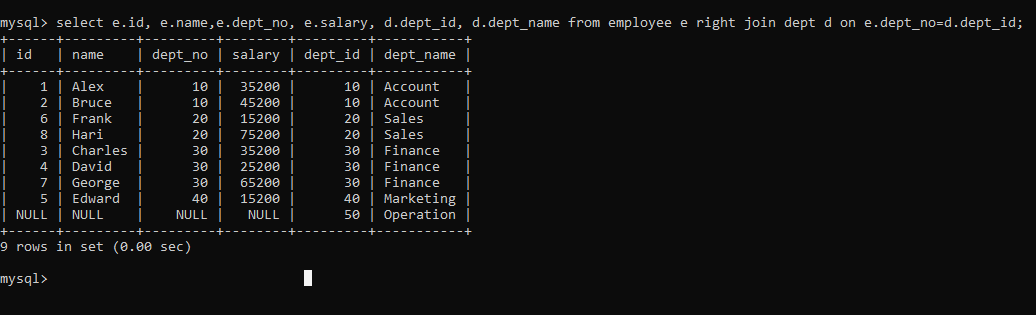


Inner join: Shows left & right tables results that matches to the condition  


Left JOIN: Shows all the records of the left table and right table will have only matching records



Right JOIN: Shows only matching records of the left table & all the records of right table



Full JOIN: It shows both left & right table records, but it is not supported in MySQL

Activity:

1. Try all the above examples
2. Using join you need to join two tables students & marks, which should give the result of students & marks along with the total marks & average marks

i.e.,

Student

|  |  |  |
| --- | --- | --- |
| RollNo | Name | Age |
| 1 | Raj | 20 |
| 2 | Ramesh | 20 |
| 3 | Ravi | 21 |
| 4 | Suraj | 22 |

Marks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mid | Physics | Maths | Chemistry | Student\_RollNo |
| 11 | 70 | 80 | 70 | 2 |
| 22 | 80 | 80 | 80 | 1 |
| 33 | 90 | 90 | 70 | 3 |
| 44 | 88 | 88 | 77 | 4 |

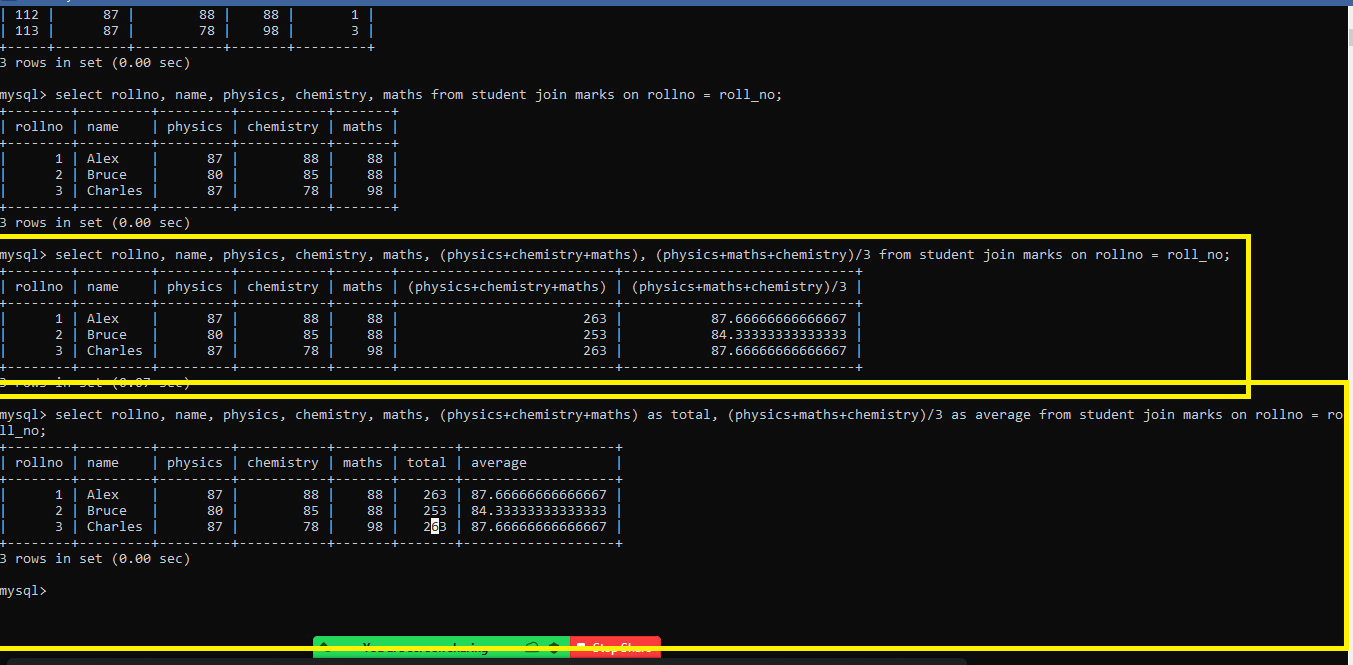
Now when you join student & marks you must get a resultant table as below;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RollNo | Name | Physics | Maths | Chemistry | Total | Average |
| 1 | Raj | 80 | 80 | 80 | 240 | 80 |
| 2 | Ramesh | 70 | 80 | 70 | 220 | 73.3 |
| 3 | Ravi | 90 | 90 | 70 | 250 | 83.3 |
| 4 | Suraj | 88 | 88 | 77 | 253 | 84.3 |

Note: The result must have column names as shown above i.e., you must get a Total, Average columns as well in the result

Note: Don’t create any table to store the result

Solution:



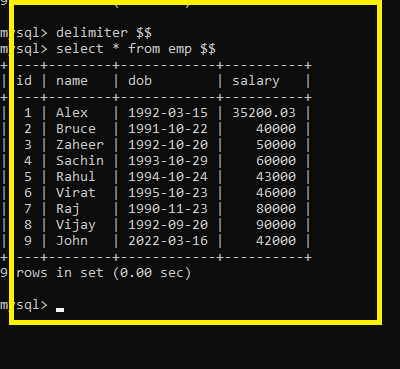
PLSQL or Stored Procedures

PL SQL means Procedural Language SQL, where you can store the procedures which will have SQL queries and other programming constructs like loops, conditions, variables etc.

Stored procedures need to be created with some names so that they can be called through that name.

Note: Since procedures will have multiple lines and each line must be terminated with ; (semicolon) we need to set some other special character as a delimiter so that MySQL doesn’t end the moment you give ;

>> delimiter $$



Syntax:

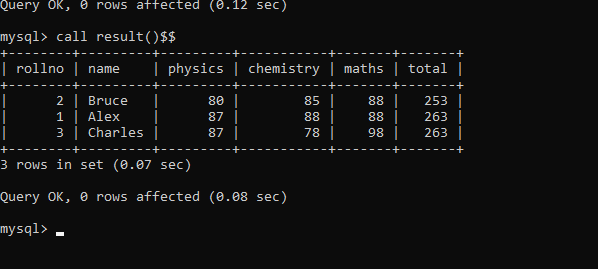
create procedure procedure\_name()  
statements  
delimiter

You can call the procedure by its name

>> call procedure\_name(args\_list)



>> call result()

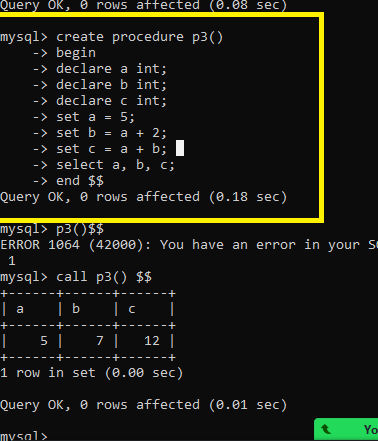


Creating variables

You can declare in the procedure through declare keyword

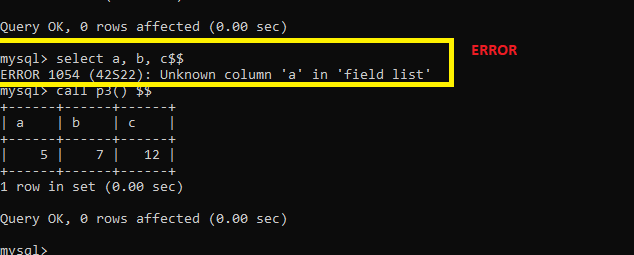
create procedure procedure\_name()  
begin  
declare variable\_name datatype;  
declare varable\_name datatype;  
…

end



set is used to assign value to the variable, once you declare variable after begin keyword, you can write multiple executable statements and using end keyword the execution block must be closed

Note: The variables you create inside procedure is not accessible outside the procedure



Global variables

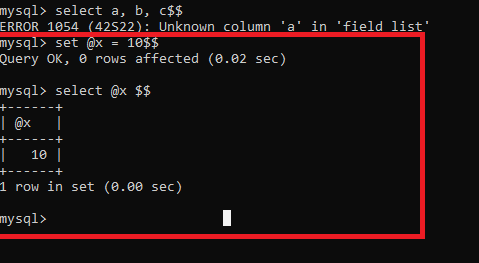
These are the variables that can be created outside the procedure and can be accessed inside/outside the procedures

>> set @variable\_name = value

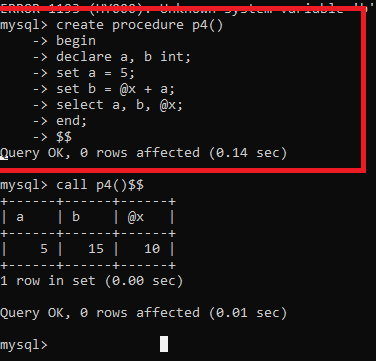
>> select @variable\_name : Here you are accessing outside

You can use @variable\_name inside procedure also

Creating a global variable without procedure



Using this variable inside some procedure



Procedures that can accept inputs i.e., parameters

In Procedures you can create 3 types of parameters

1. IN
2. OUT
3. INOUT

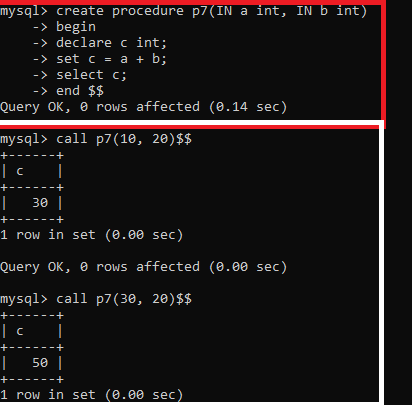
IN: Would accept the input from the caller

OUT: Would return the data to the caller

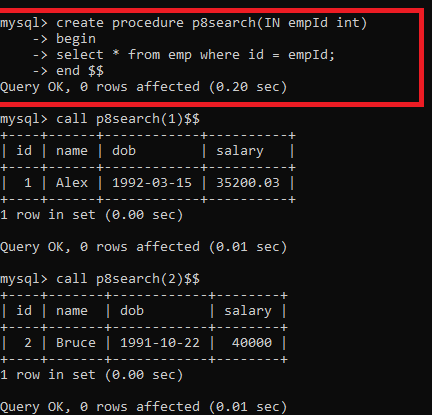
INOUT: Would accept the input from the caller also can return the data to the caller

Syntax for IN parameter

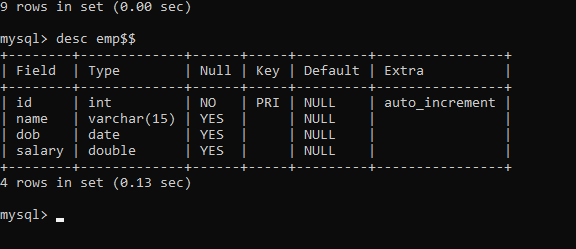
create procecdure procedure\_name(IN variable\_name datatype)  
….



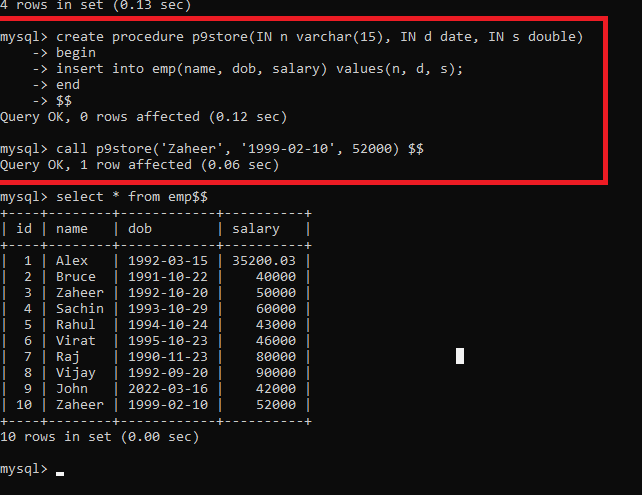
We can also use the parameter to search the records in the table



You can also insert the records by passing parameters

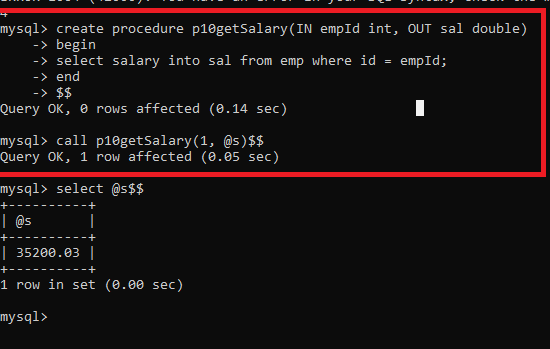


Since id is auto generated, we can create procedure to accept name, dob & salary



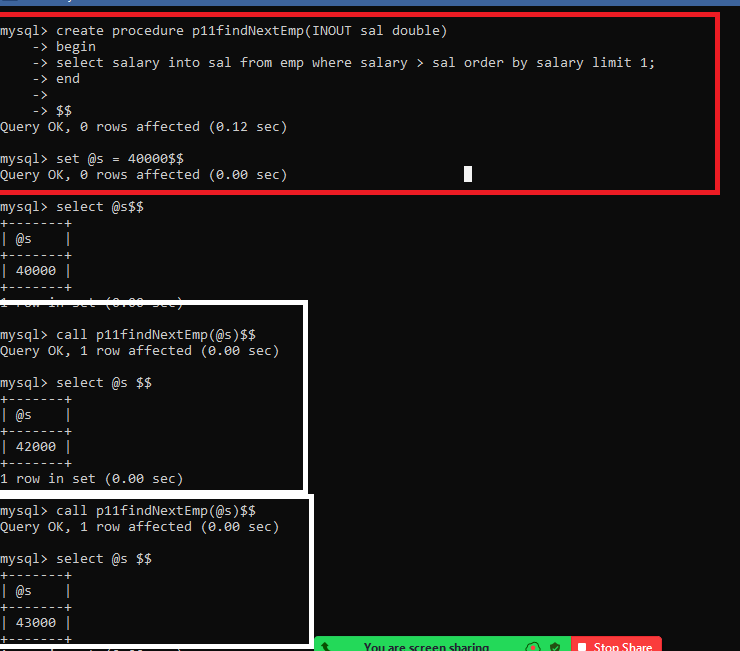
OUT parameter:

This is a parameter which can write value to a variable

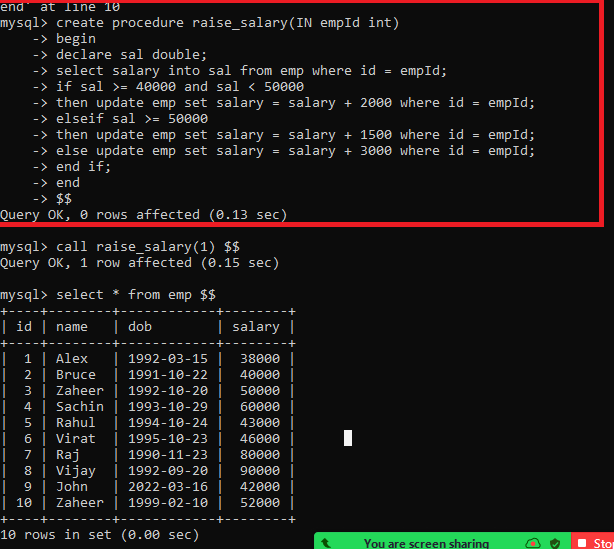


INOUT parameter

It is used when you have a parameter which can take input and also write output to a variable



Conditional statement: In MySQL you can write if, if else, if else elseif to apply conditions to execute some statements

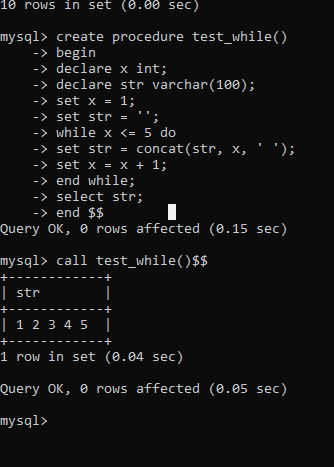


Loops

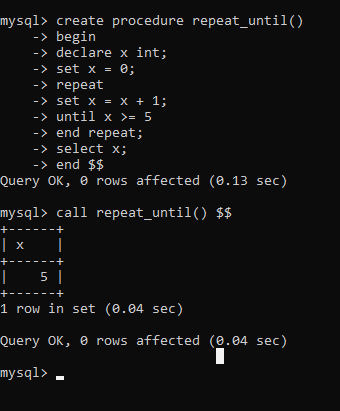
In MySQL there are 3 types of loops

1. while: top testing loop
2. repeat until: bottom testing loop
3. loop: a basic

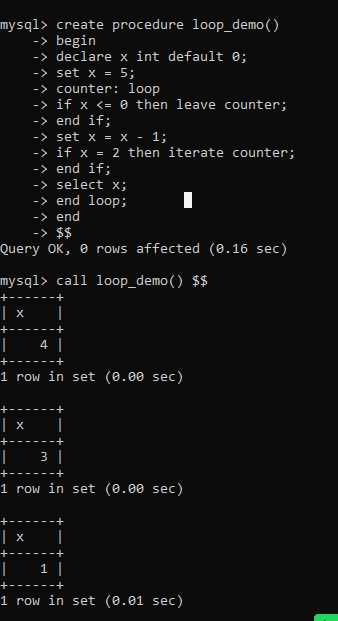
While loop



Repeat until: It is a bottom testing loop, which executes the logic then tests the condition.



Basic Loop



Triggers:

These are the programs which are automatically executed whenever some change happens in the database, whenever some rows or affected you can update some table automatically

syntax:

create trigger trigger\_name

after | before

update | insert | delete on table\_name for each row

begin

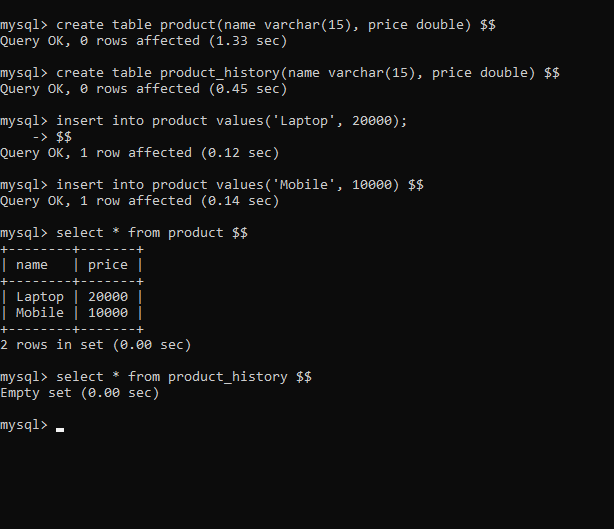
stmts

end.

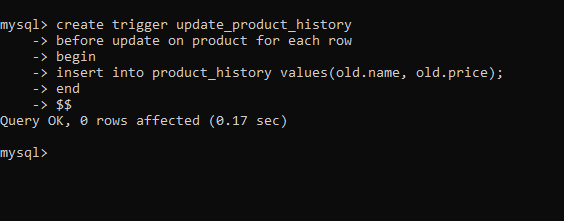
Lets create 2 tables named

products & product\_history

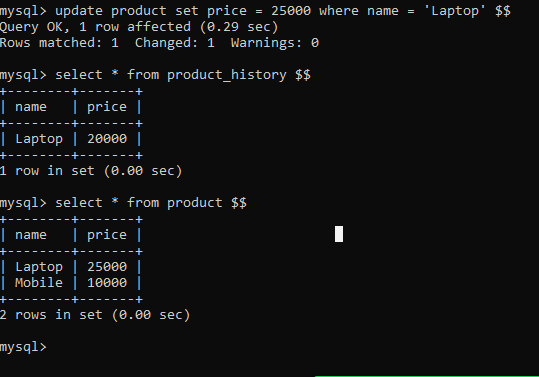
products table will have list of products and if any products price changes then the old value would be stored in product\_history.



Now you can create triggers to update the product\_history when product table price changes



Here the trigger would execute and insert records to product\_history by taking old name & price value of product table



Activity:

1. Try the above examples
2. Create a person table with 3 columns id, name and gender, store few records having genders like male and female

Create a procedure which displays the perform information based on the id, but displays with salutation i.e., if gender is male it must have Mr. beside the name & if gender is female it must have Ms. beside the name

ex: If person has below records

|  |  |  |
| --- | --- | --- |
| ID | NAME | GENDER |
| 10 | Raj | Male |
| 11 | Alia | Female |

When you call the procedure by passing 10 then it must show Mr. Raj

When you pass 11 then it must show Ms. Alia

1. Using triggers update the table with the time of modification in the history table, create 2 tables user and user\_history, whenever the user table is updated then the user\_history must have old value along with that the modified time

ex: If user table has below information

|  |  |  |
| --- | --- | --- |
| userId | name | password |
| 1 | Alex | alex123 |
| 2 | Bruce | bruce123 |

If you update Alex password to alex007 at 10:30:22, then the user\_history must have below entry

|  |  |  |  |
| --- | --- | --- | --- |
| userid | name | old\_password | time |
| 1 | Alex | alex123 | 10:30:22 |