**1. SQL Statements - 1**

(a) Writing basic SQL SELECT Statement.

SELECT Query

SELECT query is used to retrieve the data from database. SELECT query never make any

change in the database.

Syntax

[SELECT column\_name1, column\_namel... from table\_name;

If we want to retrieve data from all the columns of a table then instead of writing all the

column name just use ‘\*’. The ‘\* ‘symbol represent all the columns.

For Example



With SELECT statement different clauses can be used to display the data as per our

requirements.

WHERE clause

WHERE clause is used to specify condition in SELECT statement while fetching records

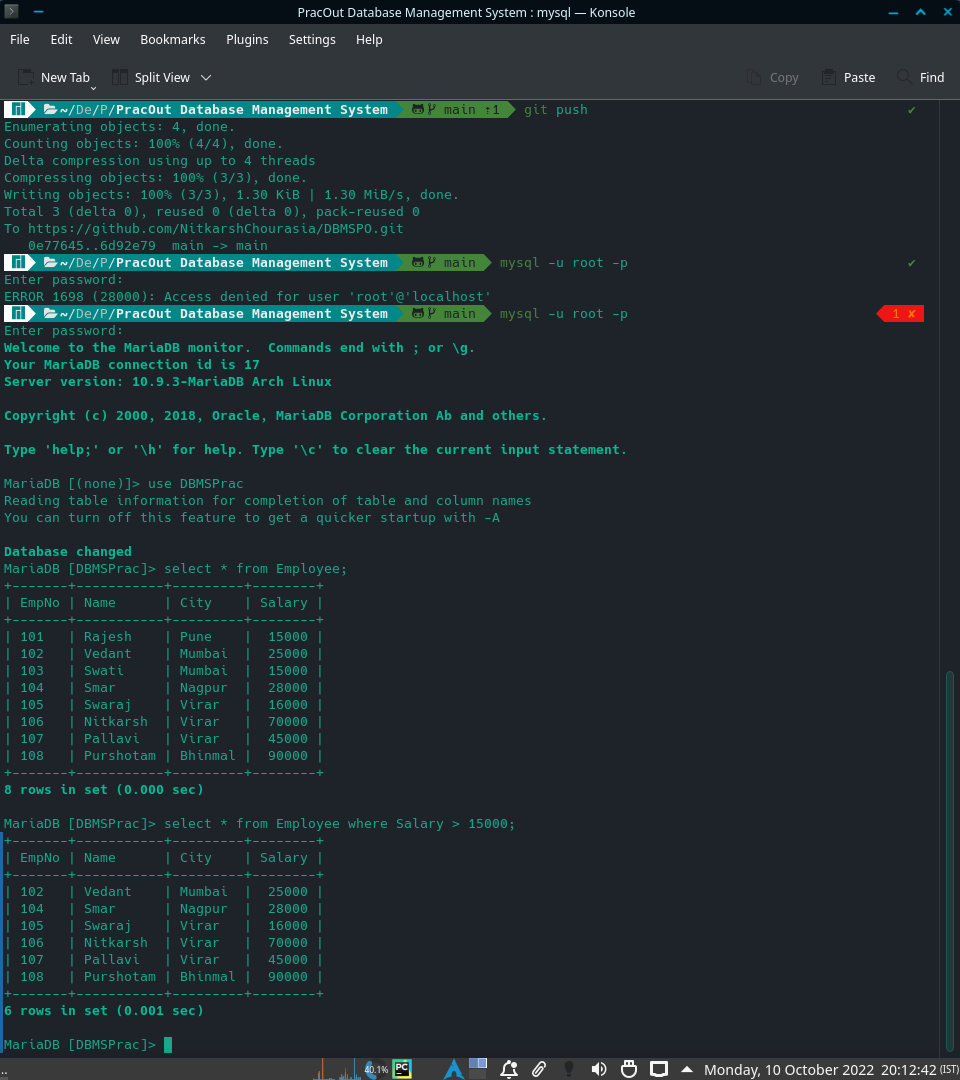
from the database. The records satisfying the condition given by where clause are retrieved.

Syntax

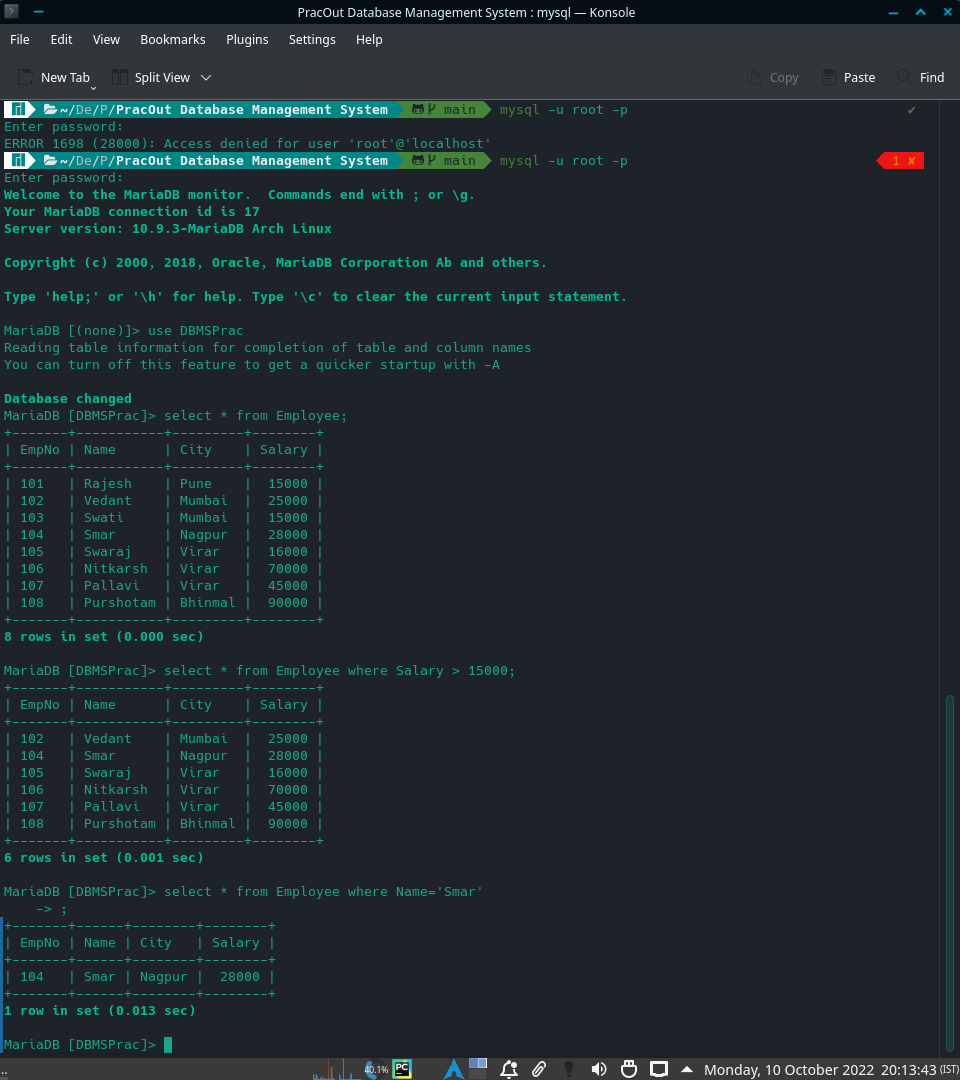
SELECT column\_1,columns\_2... from table\_name where condition;

For Example

select \* from Employee where Salary > 15000;



Select \* from Employee where Name='Smar ;



b) Restricting and Sorting Data.

DISTINCT clause

This clause is used to avoid selection of duplicate rows.

Consider a situation where duplicate values for Salary column are present in Employee

table, and the manager wants to know the salary amount given to the employees. To avoid

duplication of salary amount we use distinct keyword as follows:

select distinct(salary) from Employee;

ORDER BY clause

To arrange the displayed rows in ascending or descending order of given column, Order

By Clause is used.

For Example

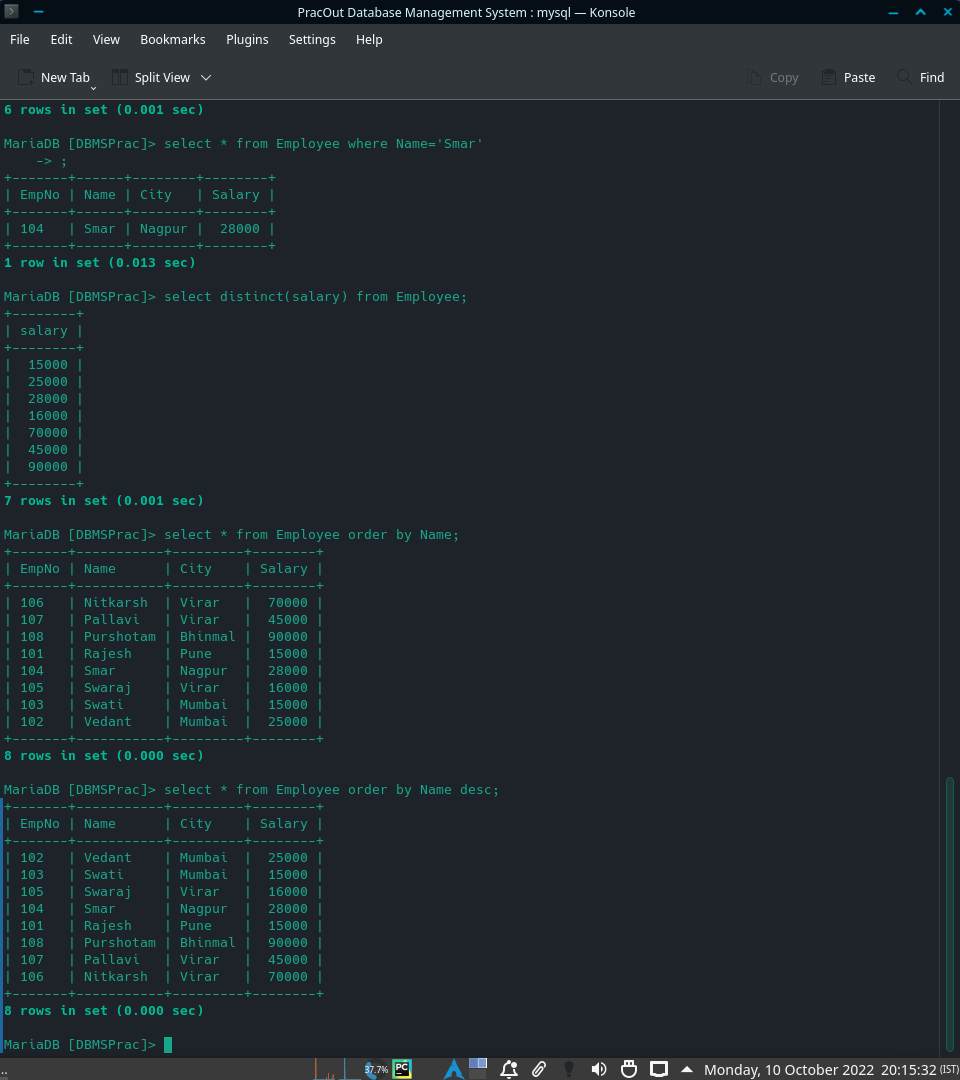
We need to display the employee information as per their joining dates in ascending order,

the query will be

Select \* from Employee order by Name;

Now to display same information in descending order on job the query will be

Select \* from Employee order by Name desc;



(c) Single - Row Functions

Single row functions can be

o Case Conversion functions : Accepts character input and returns a character value.

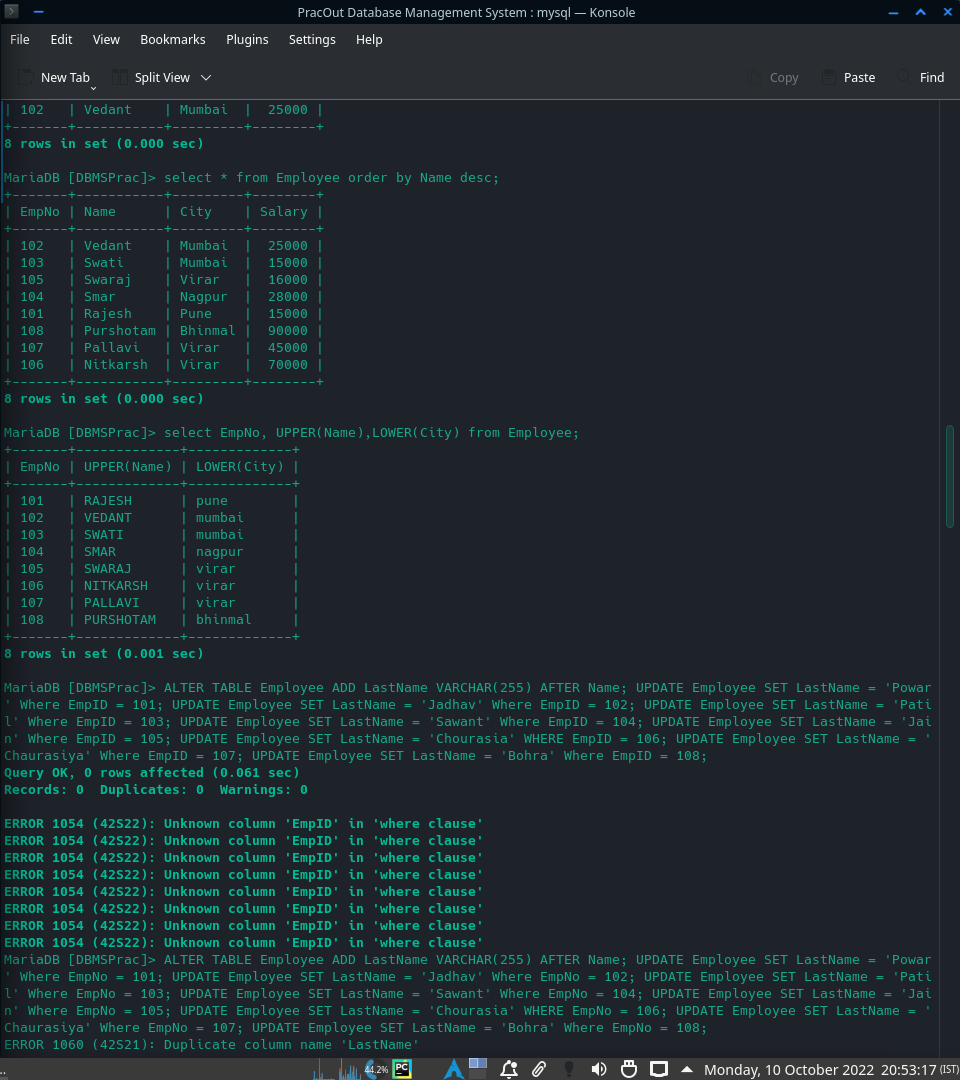
Functions under the category are UPPER, LOWER and INITCAP.

o UPPER: UPPER function converts a string to upper case.

o LOWER : LOWER function convert a string to lower case.

o INITCAP function converts only the initial alphabets of a string to upper

For example



Select Emp\_no, UPPER(Name), LOWER(city) from employees;

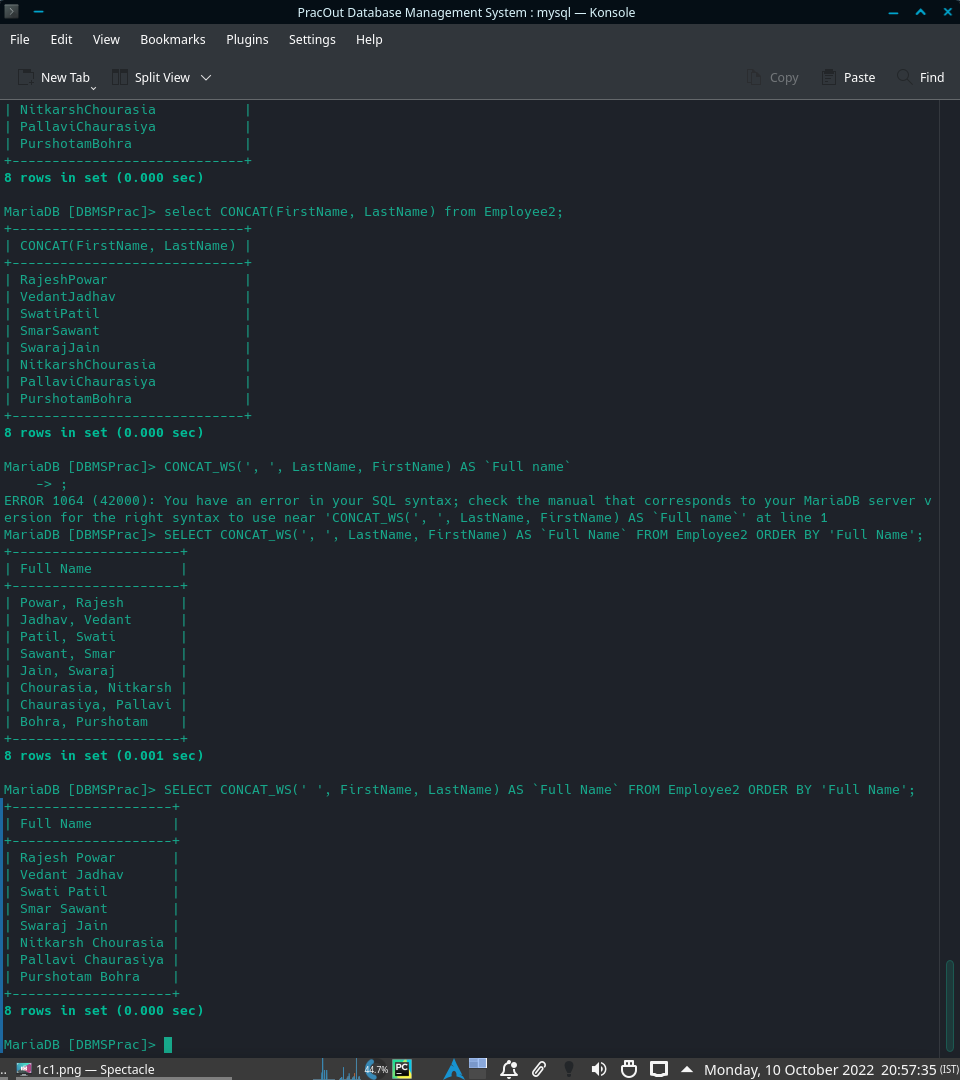
o Character functions : Accepts character input and returns number or character

value. Character functions are as follows:

o CONCAT : CONCAT function concatenates two string values.

For example

Select CONCAT(First\_name, Last\_name) from Employee2;



o LENGTH: LENGTH function returns the length of the input string

For example

Select First\_name, cesT CHP srame)f from Employee2;

o SUBSTR : SUBSTR function returns a portion of a string from a given start point to

an end point.

o INSTR : INSTR function returns numeric position of a character or a string in a

given string.

For example

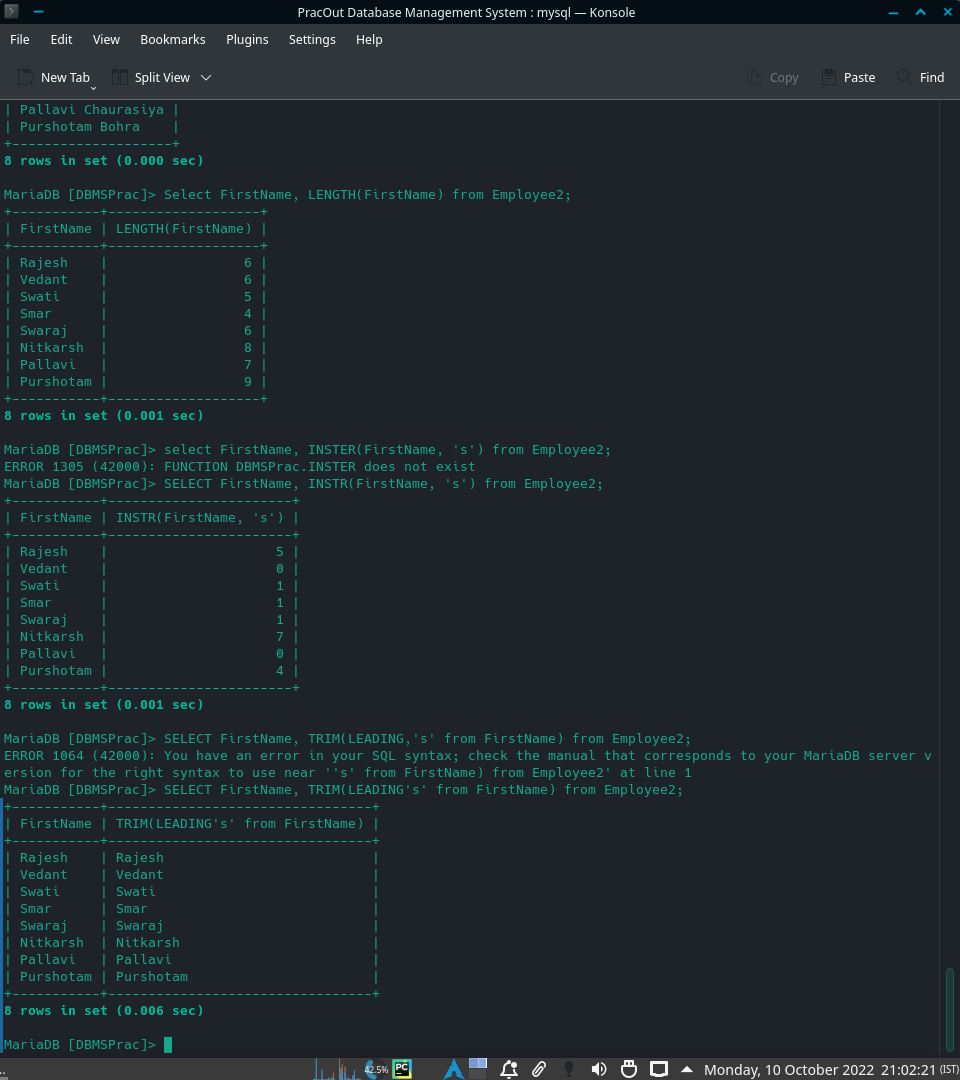
select First\_name, INSTER(First. name, ‘s

o LPAD and RPAD functions pad the given string upto a specific length with a given

character. |

o TRIM: TRIM function trims the string input from the start or end,

For example



o REPLACE : REPLACE function os Selnes characters from the input string with a

given character.

o Number functions : Accepts numeric input and returns numeric values. Functions

under the category are ROUND, TRUNC, and MOD.

o ROUND and TRUNC functions are used to round and truncate the number value.

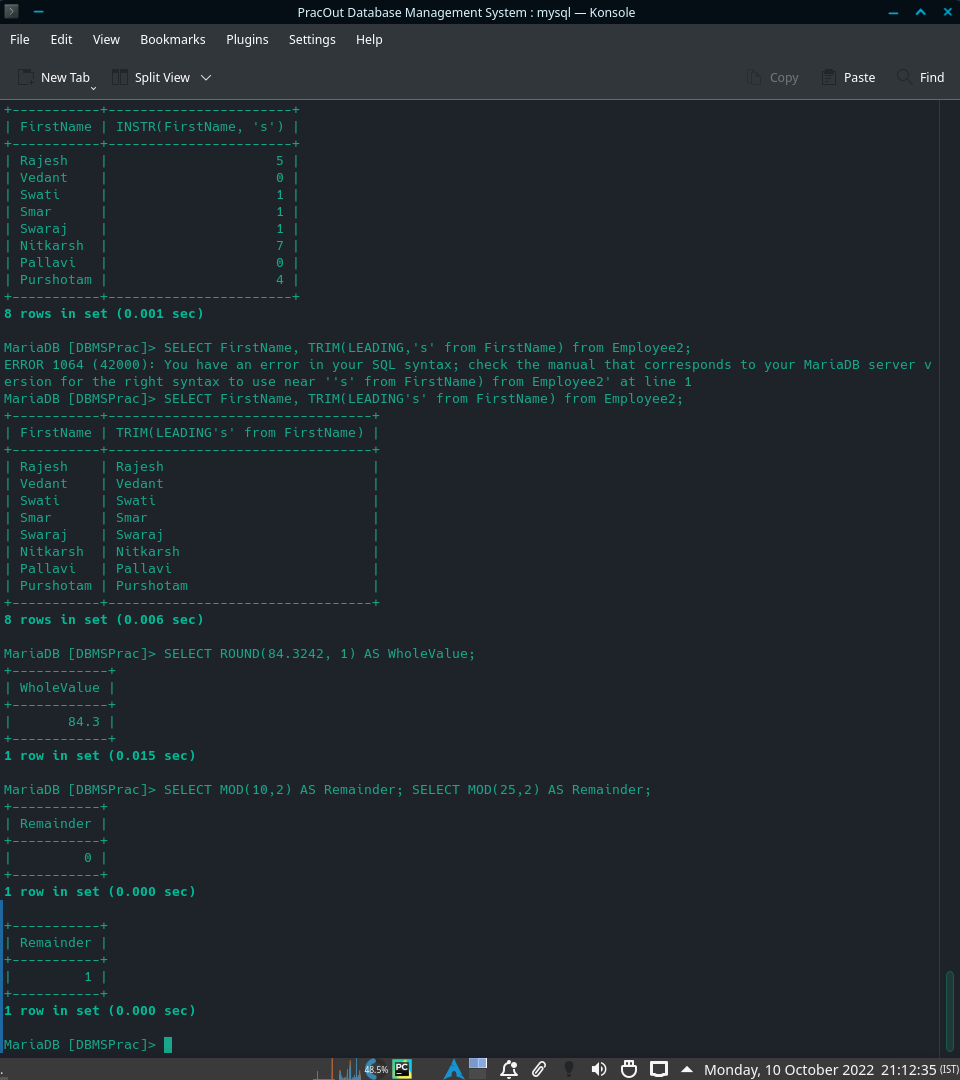
For example

SELECT ROUND(84.555,1) from dual;

© MOD is used to return the remainder of the division operation between two numbers.

For example

SELECT MOD(10,2) from dual;



**2. SQL Statements - 2**

(a) Displaying Data from Multiple Tables.

To understand joins consider following two tables.

For example

inner Join (Equi Join)

The INNER JOIN is used to display records that have matching values in both tables.

Syntax

Select column\_name\_list from table\_1

INNER JOIN table\_2 |

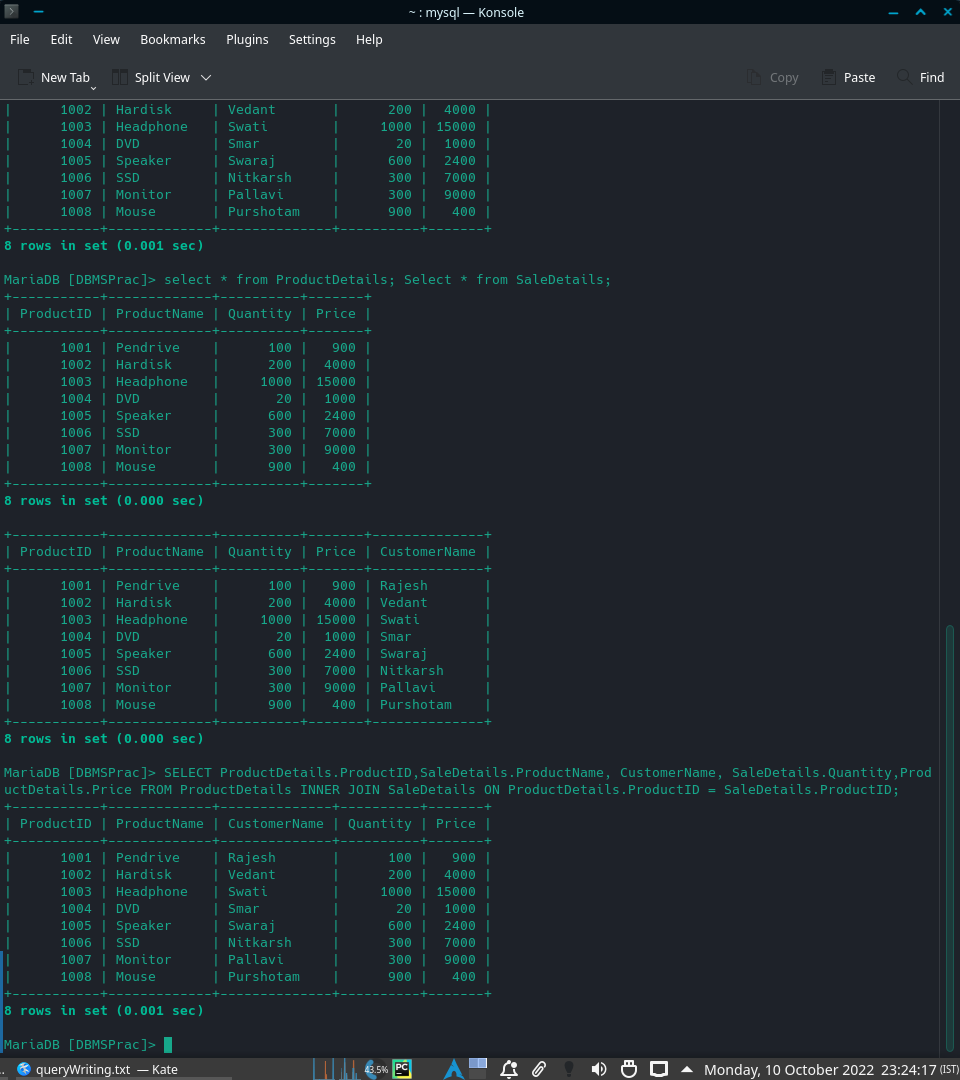
where table\_1.column\_name = table\_2.column\_name

For Example

select product\_details.product\_id, product\_name, customer\_name , sale\_details.quantity,

product\_details.price from product\_details INNER JOIN sale\_details on —

product\_details.product\_id=sale\_details. product\_id ;



Outer Join -

Outer Join is based on both matched and unmatched data. Outer Joins subdivide further

into,

(i) Left Outer Join (ii) Right Outer Join

The SQL LEFT JOIN returns all rows from the left table, even if there are no matches in

the right table, Null values are shown at the place of right table values.

Syntax

SELECT column-name-list

from table-name1

LEFT OUTER JOIN

table-name2

on table- -namel. column-name = ‘table- name2. column- -name;

For Example

SELECT product\_details.product\_id, product’. details. product\_name,

sale\_details.customer\_name FROM product\_details LEFT OUTER JOIN sale. details ON.

sale\_details. product\_id =product\_detdils.product\_id:

Jails. customer\_name FROM product\_details LEFT OUTER JOIN sale\_details ON saledet

jails. profuctid oye ees. product\_id;

(ii) Right Outer Join

Returns all rows from the right table even if there are no matches in the left table. Null

values are shown at the place of left table values.

Syntax

select column-name-list from table-name1

RIGHT OUTER JOIN

table-name2

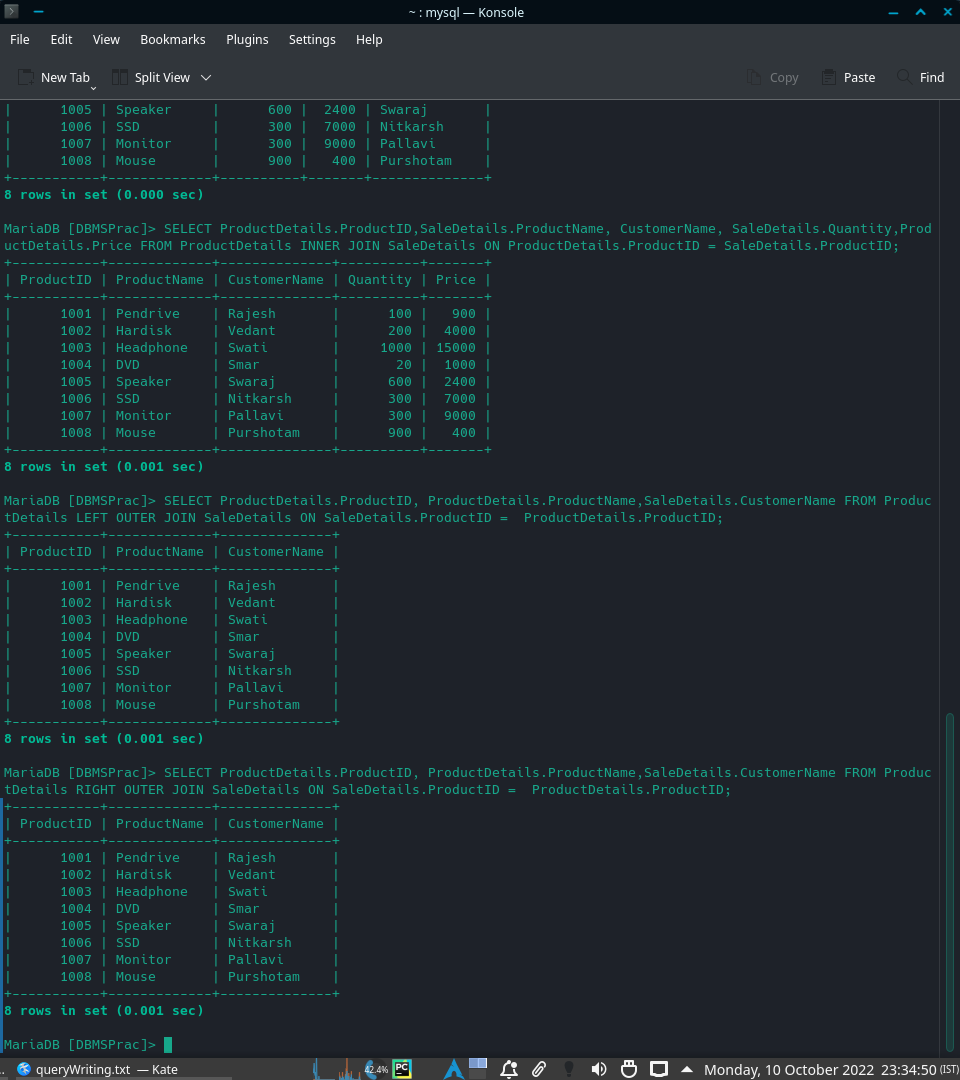
pon table-name1.column-name = table-name2.column-name;

For Example

SELECT product\_details.product\_id, product\_details.product\_name,

sale\_details.customer\_name FROM product\_details RIGHT OUTER JOIN sale\_details ON

sale\_details.product\_id =product\_details. product\_id; a



(b) Aggregating Data Using Group Functions.

Count()

This function returns total number of values of specified column of the table.

For Example

Write a query to retrieve count of employees who join in 2015 year.

‘Select ‘count(Emp.no):f from. Employee2 where First\_name like 'swa%';

Sum(Q)

This function returns sum of all the values of specified column of the table.

For Example

Write a query to to find total salary amount paid to all the employees.

select ct Sum Salary) from ployed

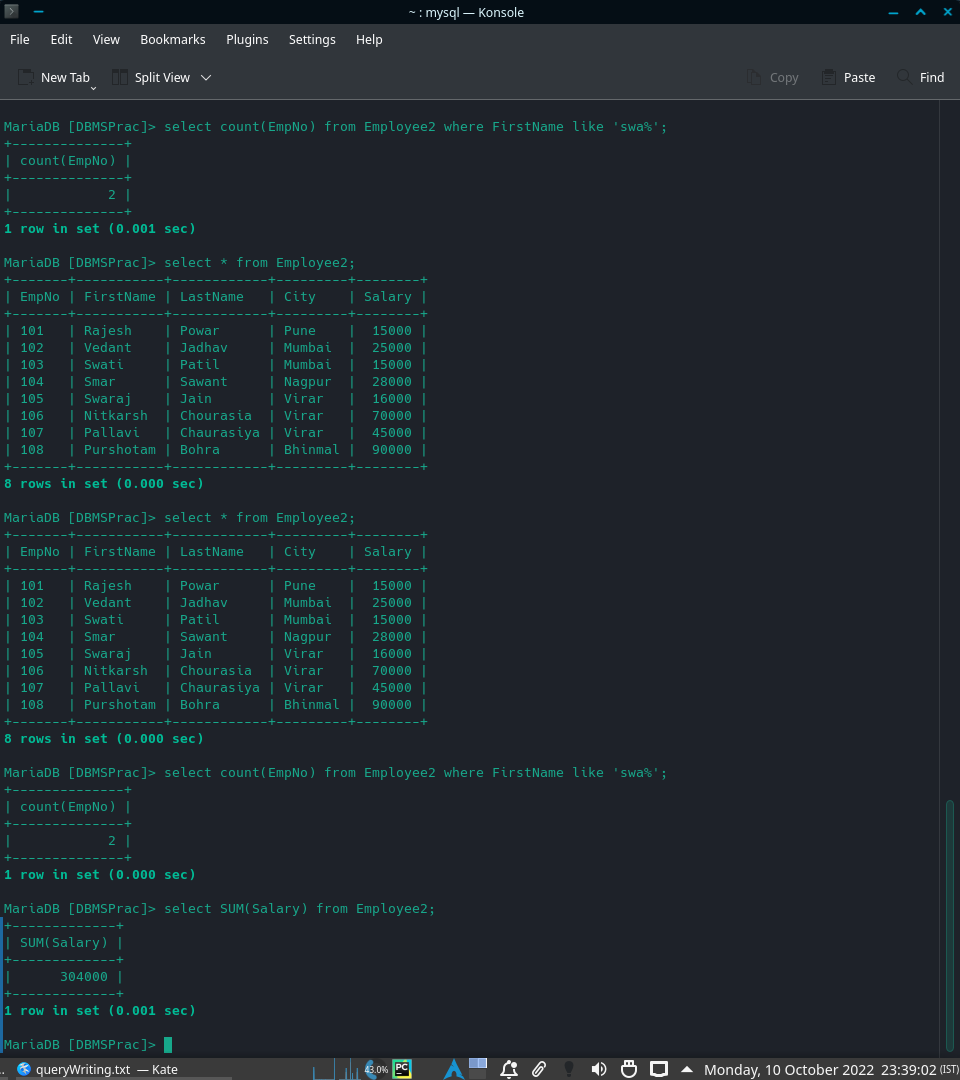
Min()

This function returns smallest value from specified column of the table.

For Example

Write a query to retrieve record of employee who gets least salary.

[select \* from Employee2 where Salary =(Select min(Salary) from Employee2); |



Max()

This function returns greatest value from specified column of the table.

For Example

Write a query to retrieve record of employee who gets maximum salary.

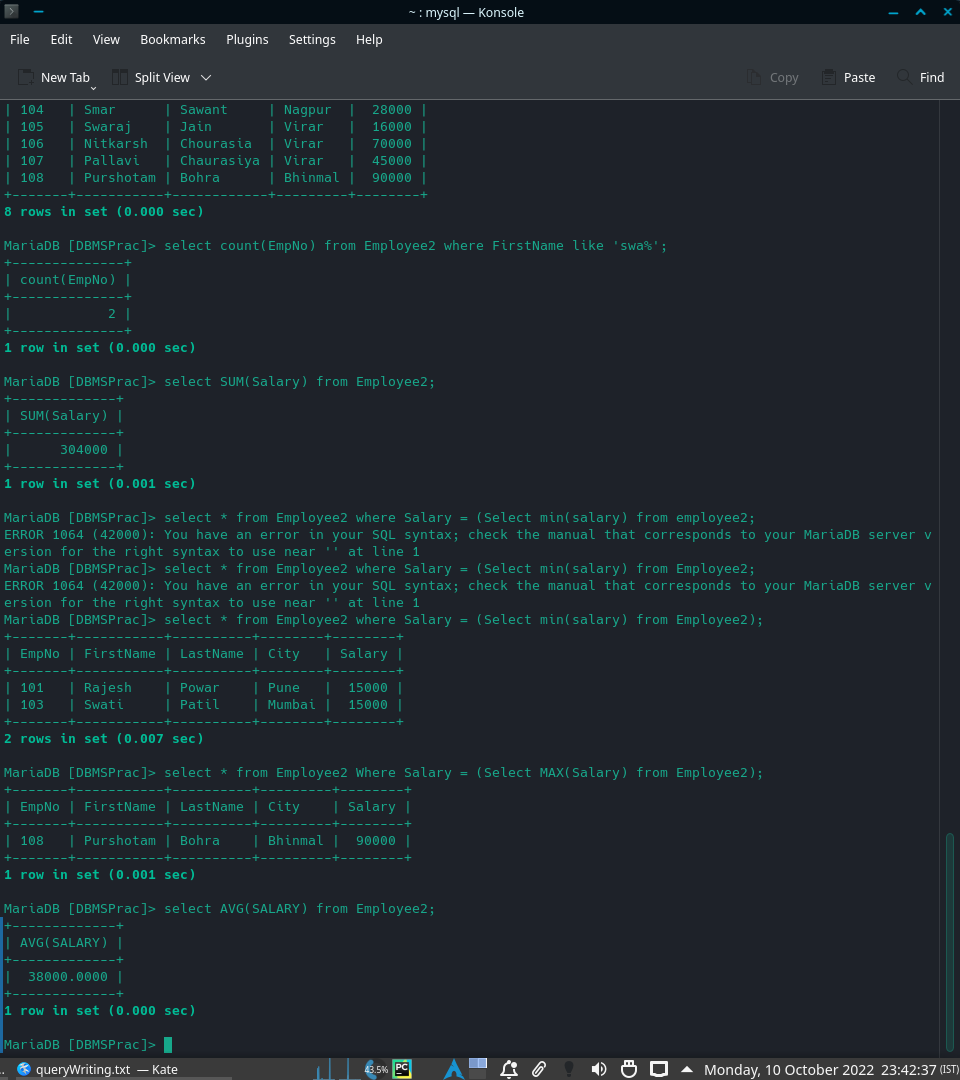
[Select \* from Employee2 where Salary=(Select max(Salary) from Employee2);

Avg()

The following SQL statement finds the average salary of all employee:

For example

select Avg(salary) from Employee2;



(c) Sub queries

Sub-Queries

Writing a query inside another query is known as nested query or subquery. The inner

query get executed first, then the output on inner query is given as input to outer query.

Consider the previous emp table

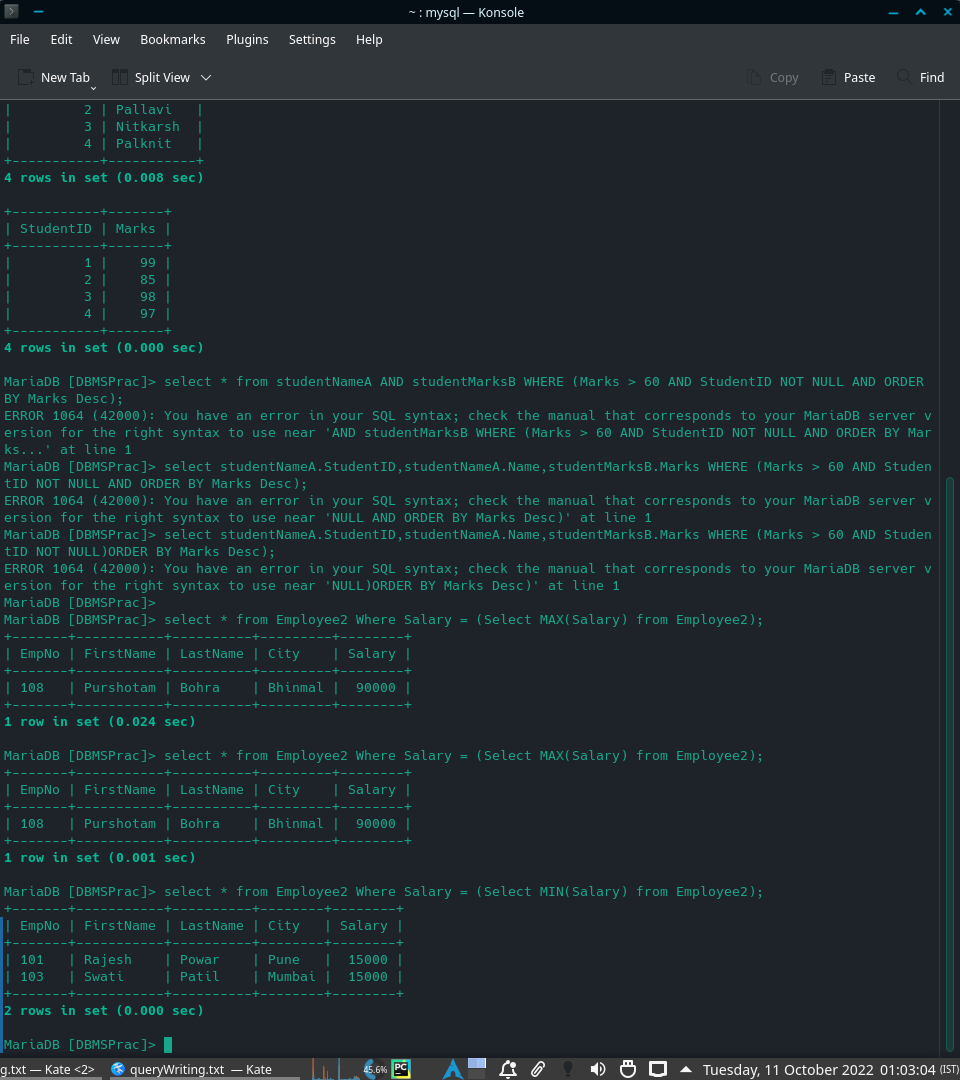
For Example

To display records of employees working in SMITH’s department

Select \* from emp where deptno =

(select deptno from emp where ename = ‘SMITH’);

Output



**3. Manipulating Data**

Data Manipulation Language (DML)

- The Data Manipulation Language (DML) is used for accessing and manipulating data in a

\_ database. It allows users to access, insert, update, and delete data from the database.

© To insert record into the table - INSERT

o Toaccess or read records from table -SELECT

© Update the records in table - UPDATE

© Delete the records from the table- DELETE

(a) Using INSERT STATEMENT

1 Inserting record into tables

After creation of table the insert command is used to insert one or more records in the

lable,

Insert query has different forms.

Format 1: Inserting a single row of data into a table

Syntax

Insert into table\_name[(column\_namet, column \_name2,--)]

values (value1, value2....);

For example

Insert 5 records in Employee table (102 Swati’, 9015-01-

Insert into Employee values(101,'Rajesh’, 1995-110 05-02-

After inserting 5 records in table the table will look like as follow\*:

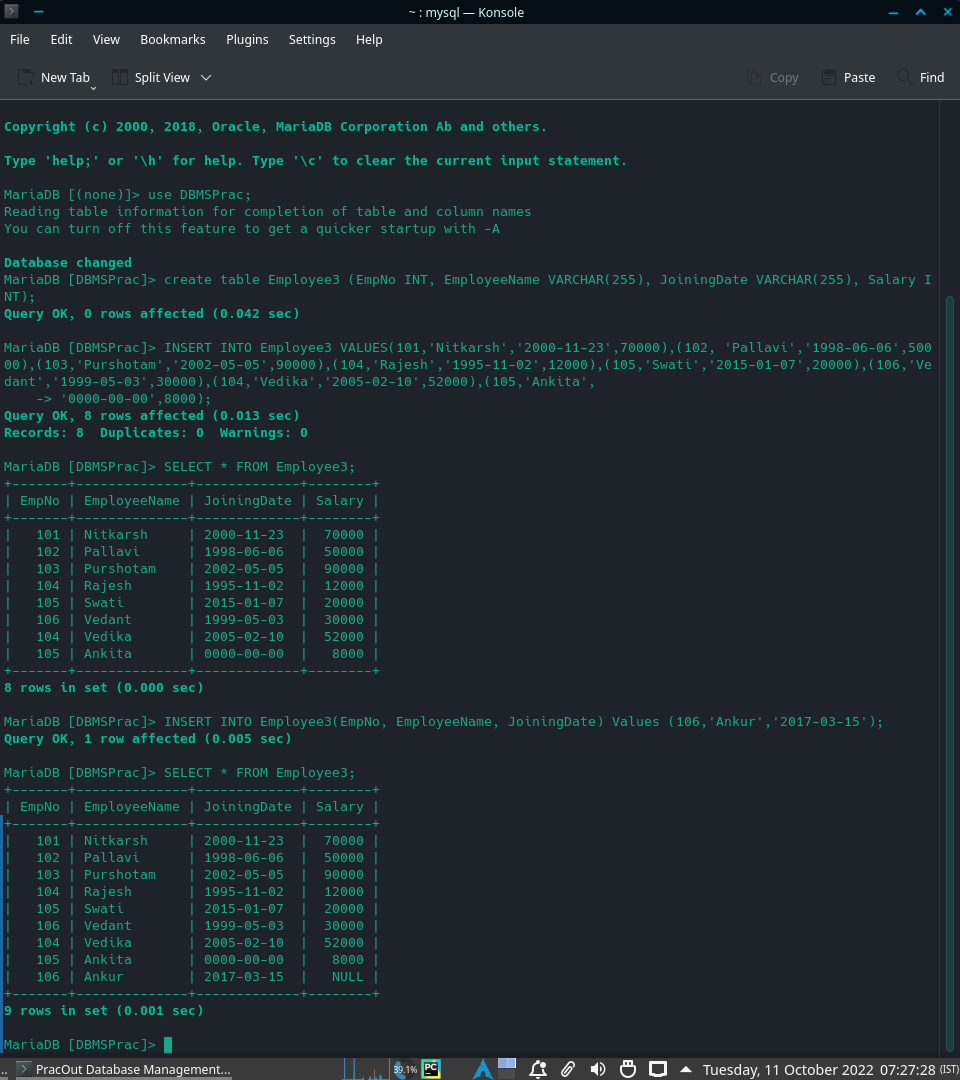
2. Consider we do not have value for salary while inserting a new record in Employee

otable. Then the query will be

Insert into Employee(Employee\_no, Employee\_name, Joini |

values(106,’Ankur,'2017-03-15'); Ingedate)

It will set salary value for the employee as NULL.



Format 2 : Inserting data into a table from another table

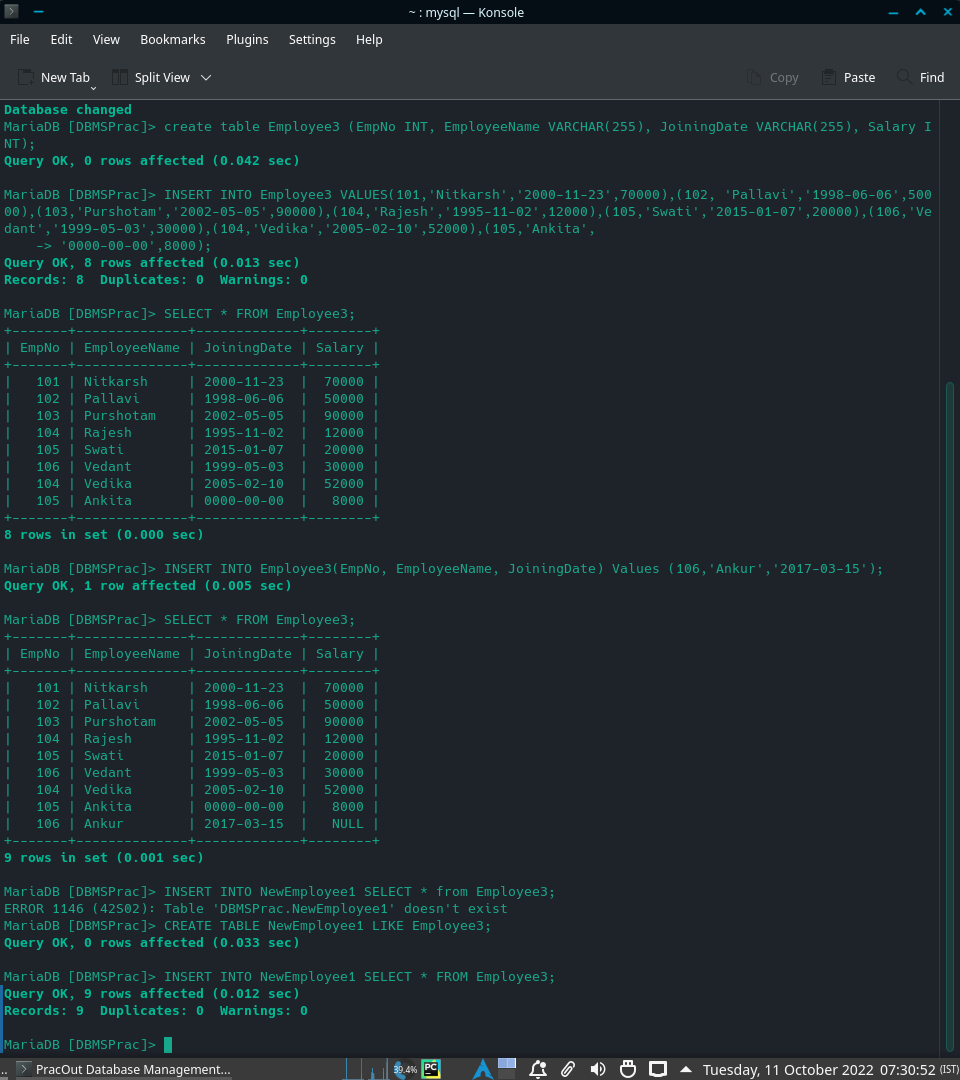
Syntax

Insert into table\_name select column\_name1, columne\_name2 from table\_name;

Insert records in newEmployeel table same as in Employee table.

For Example

Insert into newEmployee1 select \* from Employee;



(b) Using DELETE STATEMENT

Delete

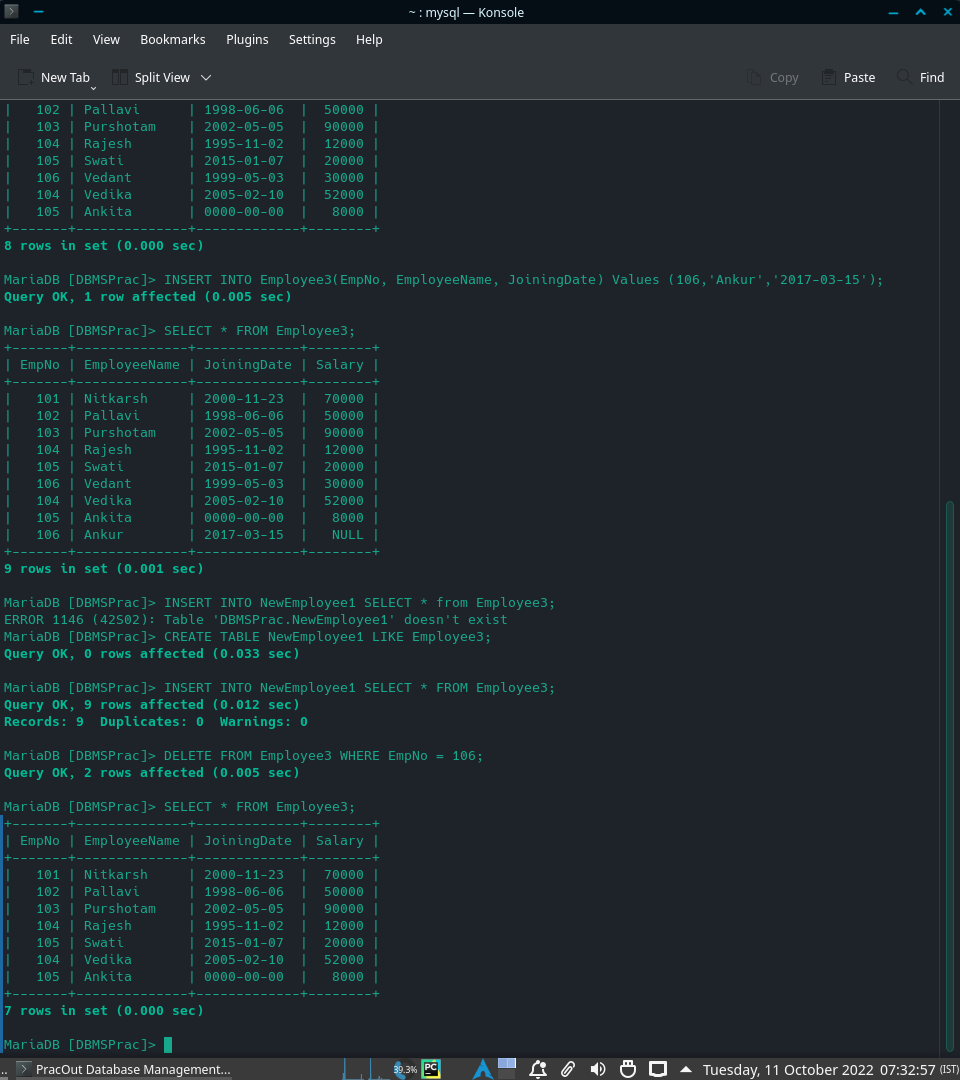
As per requirement, the records from existing table can be removed using delete

command. Delete command can have ‘WHERE’ clause optionally.

Syntax

Delete from table\_name;

For Example



Write a query to remove record of Employee\_no 106 from Employee table,

Delete from Employee where Employee\_no = 106;

(c) Using UPDATE STATEMENT

Update

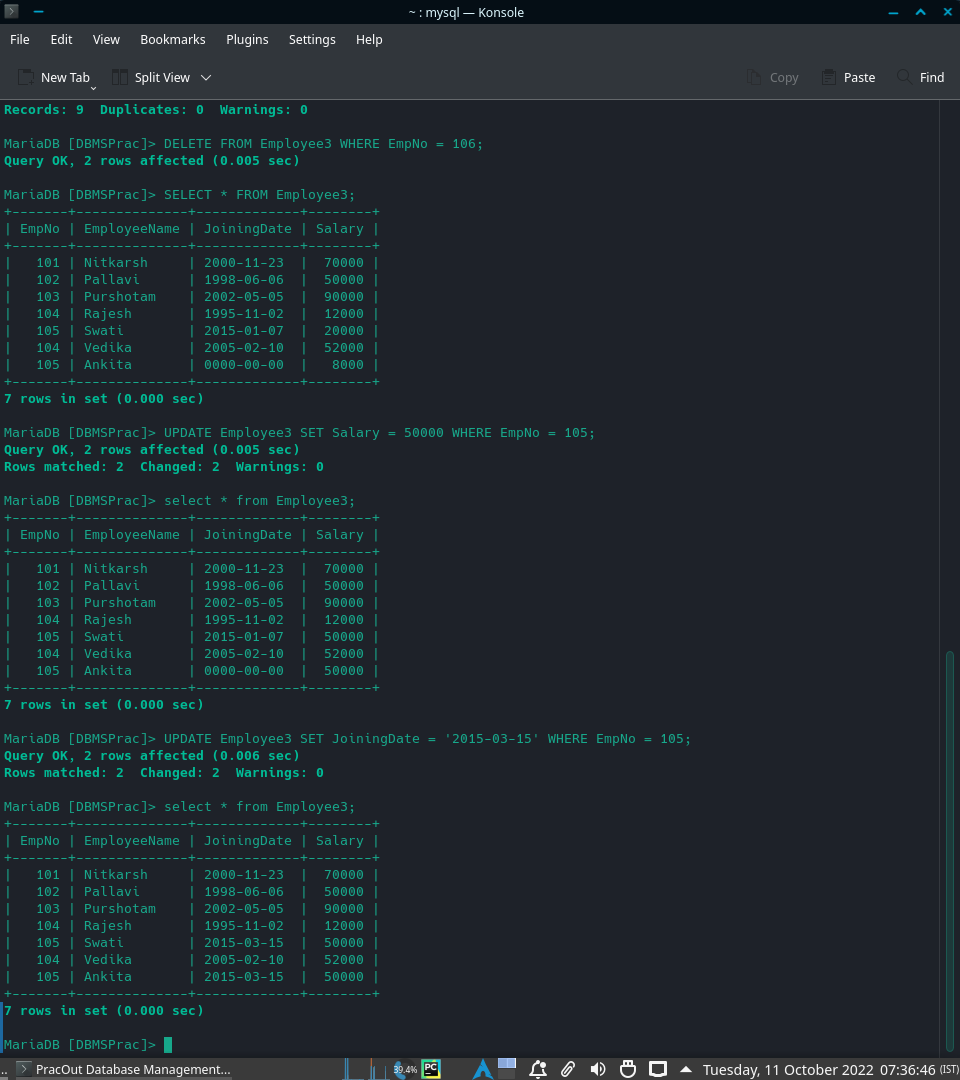
To make changes in the database ‘update’ command is used. The update command

consists of ‘set’ clause and an optional ‘where’ clause’. ‘WHERE? clause is used to make

changes in specific records.

Syntax

Update table\_name set column\_name = new\_value [where condition];



**4. Creating and Managing Tables**

(a) Creating and Managing Tables

Creating Table

The CREATE TABLE statement is used to create table in database.

Syntax .

CREATE TABLE table\_name (

column] datatype[size],

column2 datatype [size],

column3 datatype[size],

For Example

Following command creates a table Employee having four columns

Employee\_name, Joining \_date, and Salary.

CREATE Table Employee ( —

Employee\_no integer(3), °

Employee\_name varchar(20),.

joining\_date date ,

Salary integer(6));

1. Creating New Table from Existing Table

Consider the existing table Employee

Creating new table same as of existing table.

Syntax

Create table-table\_ name as select \* from existing table name;

For Example

Create table nEwEmpleyess ; RES

PAS ee

select \* -from Erployessi:

Employee table.

Creating new table having specific fields but all the records from existing table.

Syntax

Create table table\_name as select field\_\,field\_2... from existing\_table\_name, a

For Example

Create table newEmployee2

As

select Employee\_no, Employee\_name from Employee;

After executing this command the newly created table will contain two fields such as

Employee\_no and Employee\_name , and also contain all the corresponding records as in

Employee table.

Creating new table having specific records but all the fields from existing table.

Syntax

[ Create table table\_name as select \* from existing \_table\_name where condition

For Example

Create table newEmployee3

As

select \* from Employee

where Salary > 80000;

The newly created table will have same structure as of employee table, but it will contain

records of only those Employees who got Salary above 80000 Rs.

Creating new table having no records but all the fields from existing table.

That means copying only structure of existing table

Syntax

Create table table\_name as select \* from existing table\_name where false condition

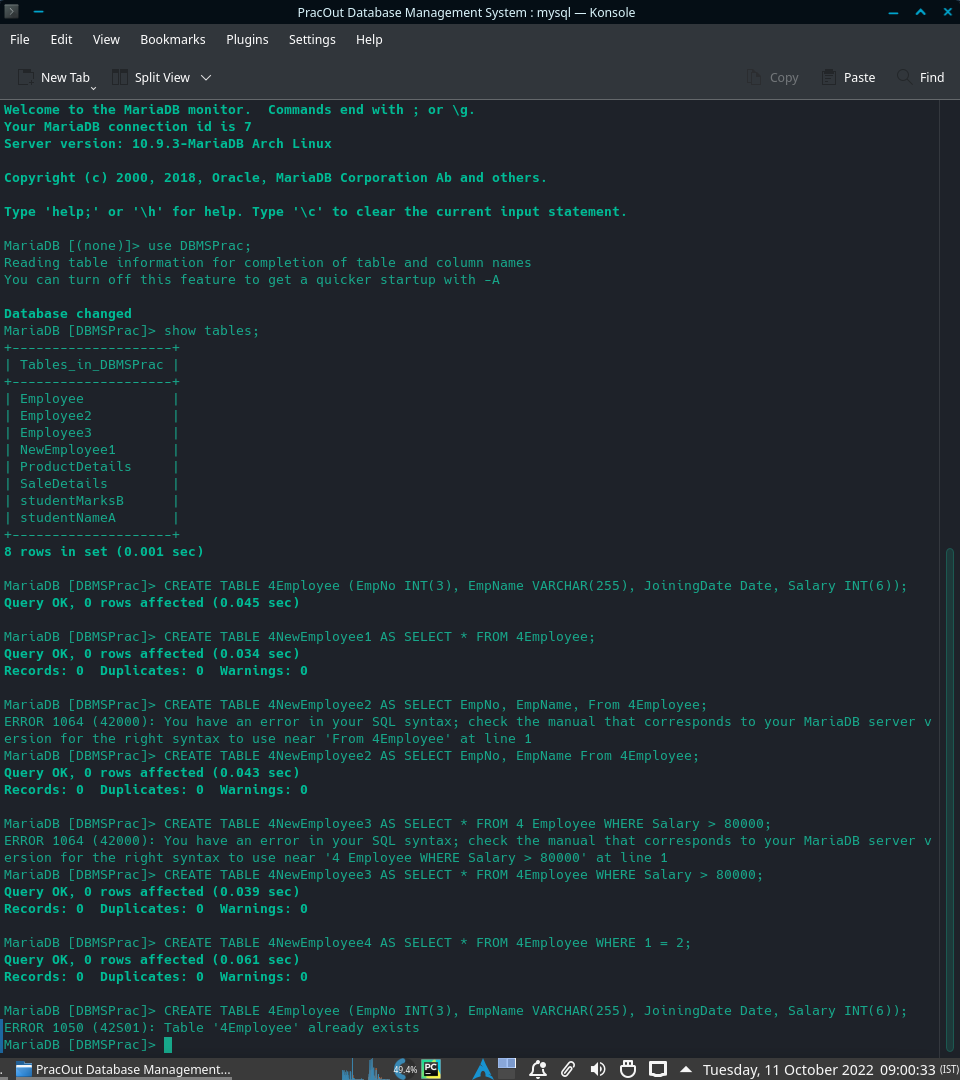
For Example

Create table newEmployee4

As

select \* from Employee

where 1=2;



Here 1=2 is the false condition

The newly create “exist iw

1ewly created table will have same structure as of existing table, but it will not copy

ANY records from,

2. Modifying Table

ALTER TABLE query is used to modify structure of a table which is already exists in the

database. We can add, delete or modify column.

Adding New Column in a Table

Syntax

ALTER TABLE table\_name ADD column\_name datatype;

For Example

ALTER TABLE Emp\_dept ADD column city varchar(20);



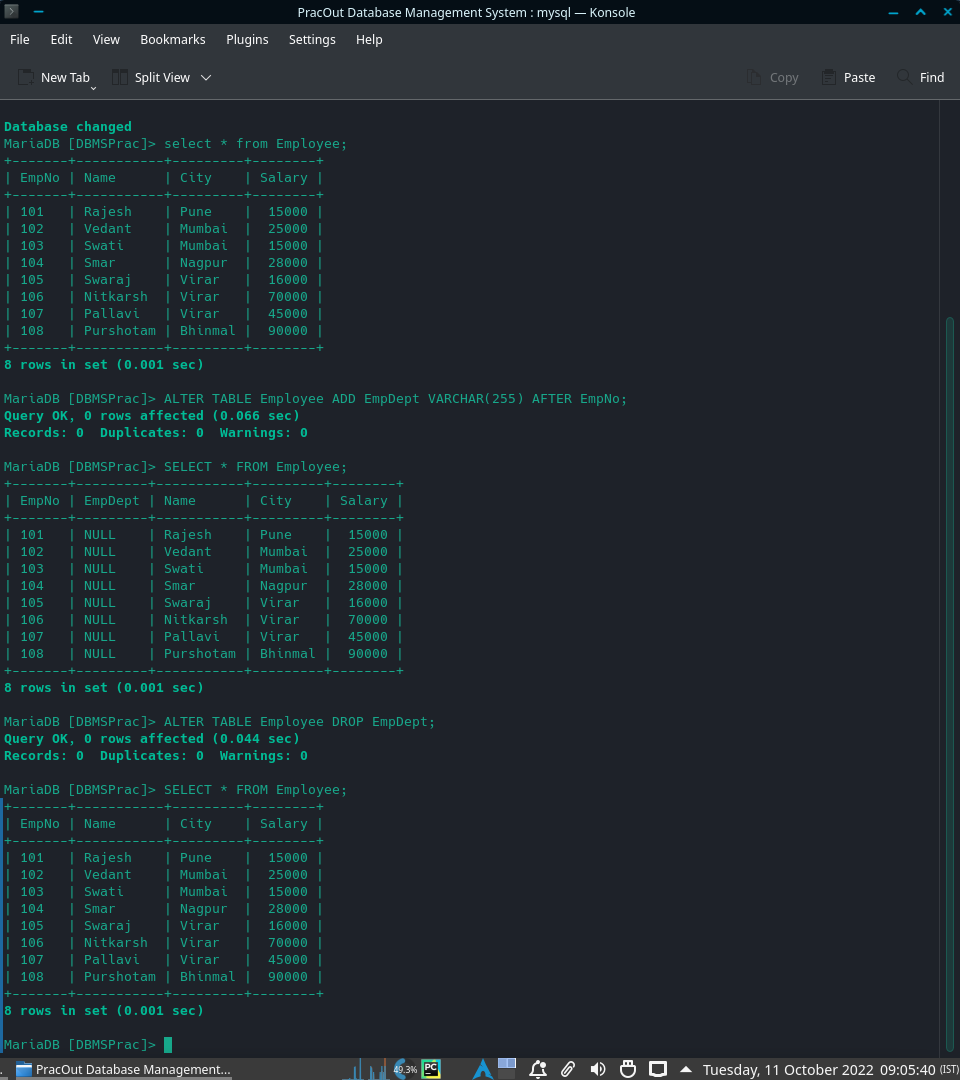
Dropping Column from Table

Syntax

ALTER TABLE table\_name DROP COLUMN column\_name;

For Example:

ALTER TABLE Emp\_dept DROP COLUMN dept;



Modifying Column of a Table

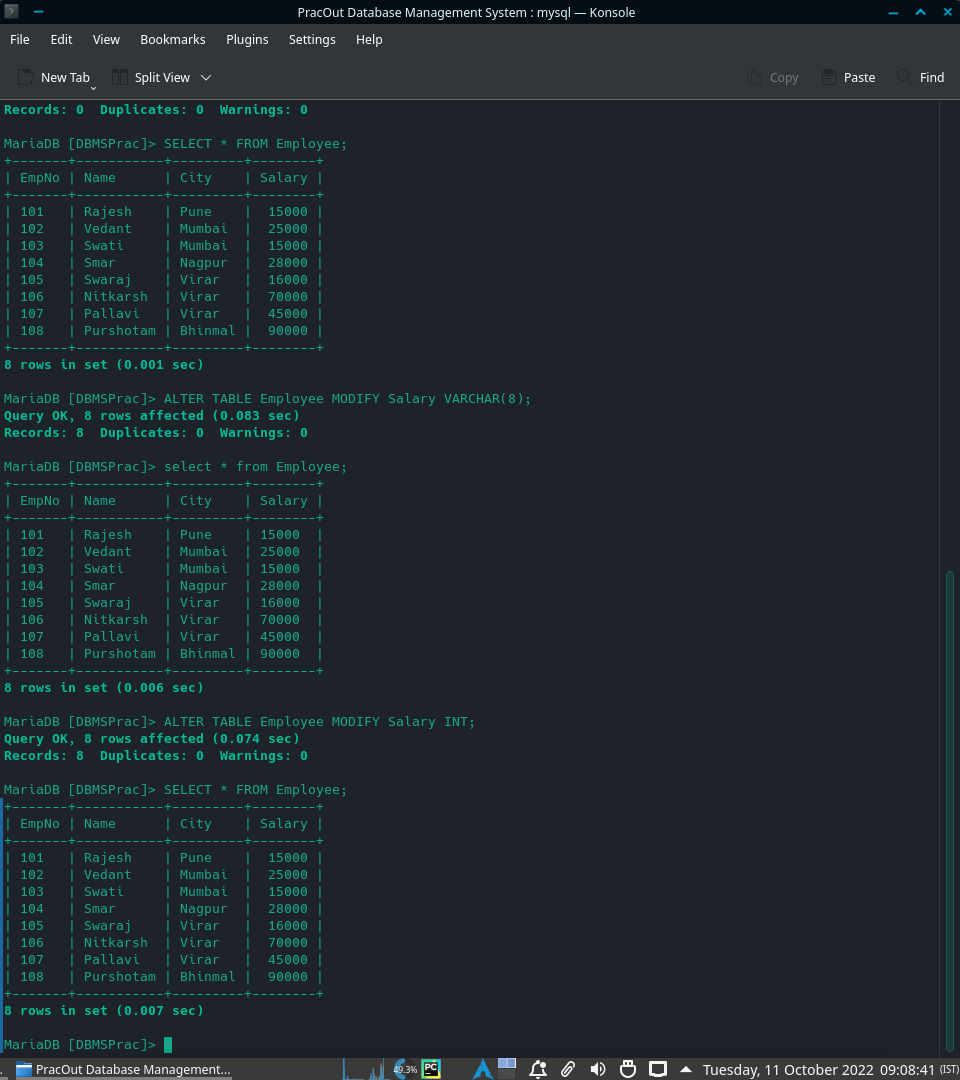
Here we are changing the data type and size of column roll\_number.

Syntax

ALTER TABLE table\_name MODIFY COLUMN column\_name data\_type;

For Example

ALTER TABLE Employee modify column Employee\_no varchar(4);



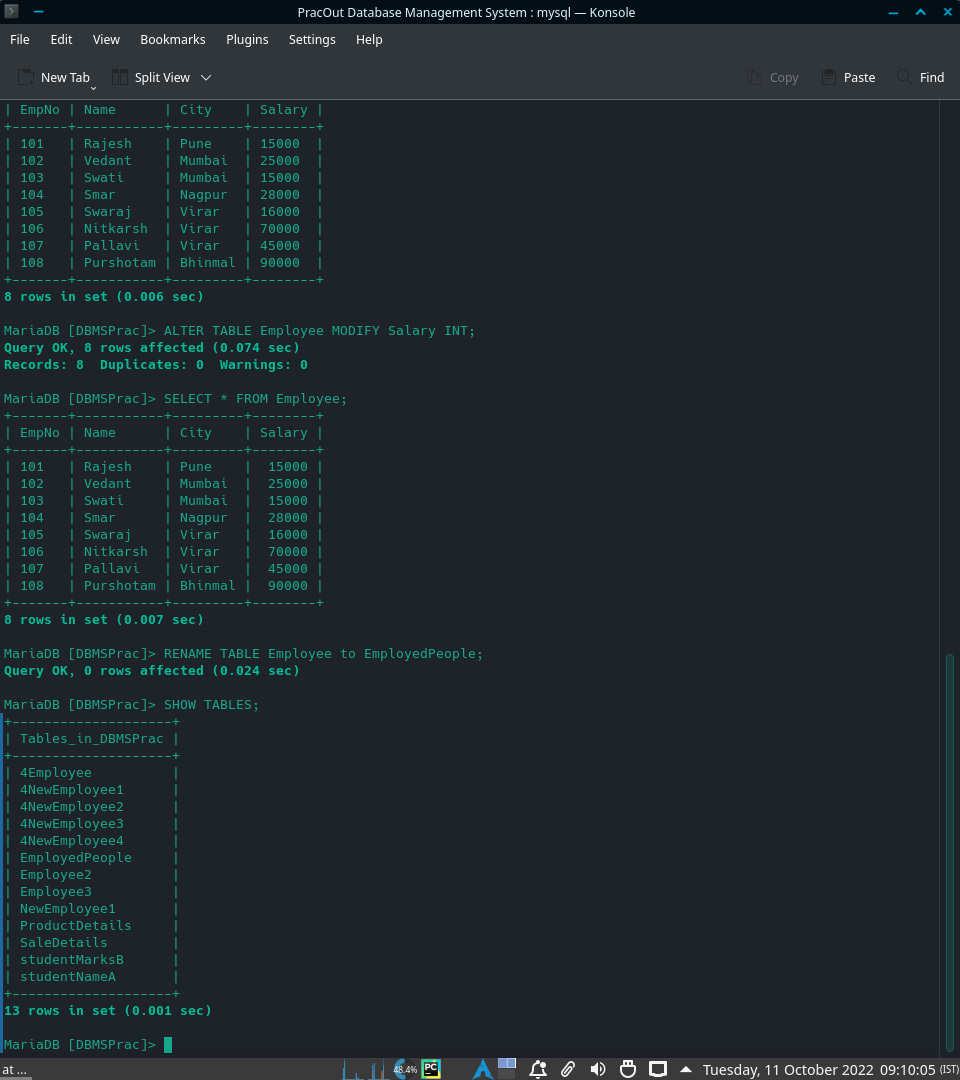
3. Renaming Table

Syntax

rename table current\_table\_name to new\_table\_name;

For Example

[rename table Employee to Emptable;



4. Deleting Table

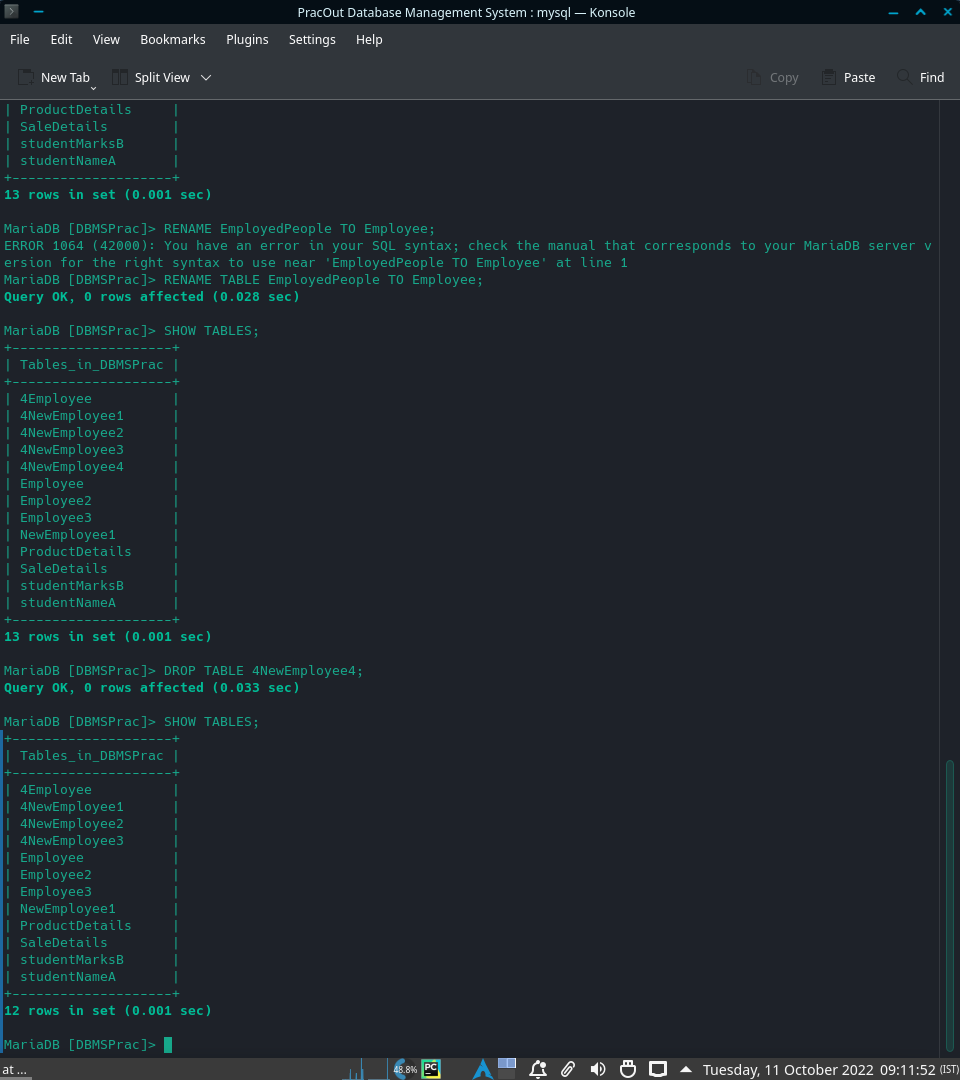
DROP TABLE query is used to delete table.

Syntax

drop table table\_name;

For Example

Drop table Emptable:



(b) Including Constraints

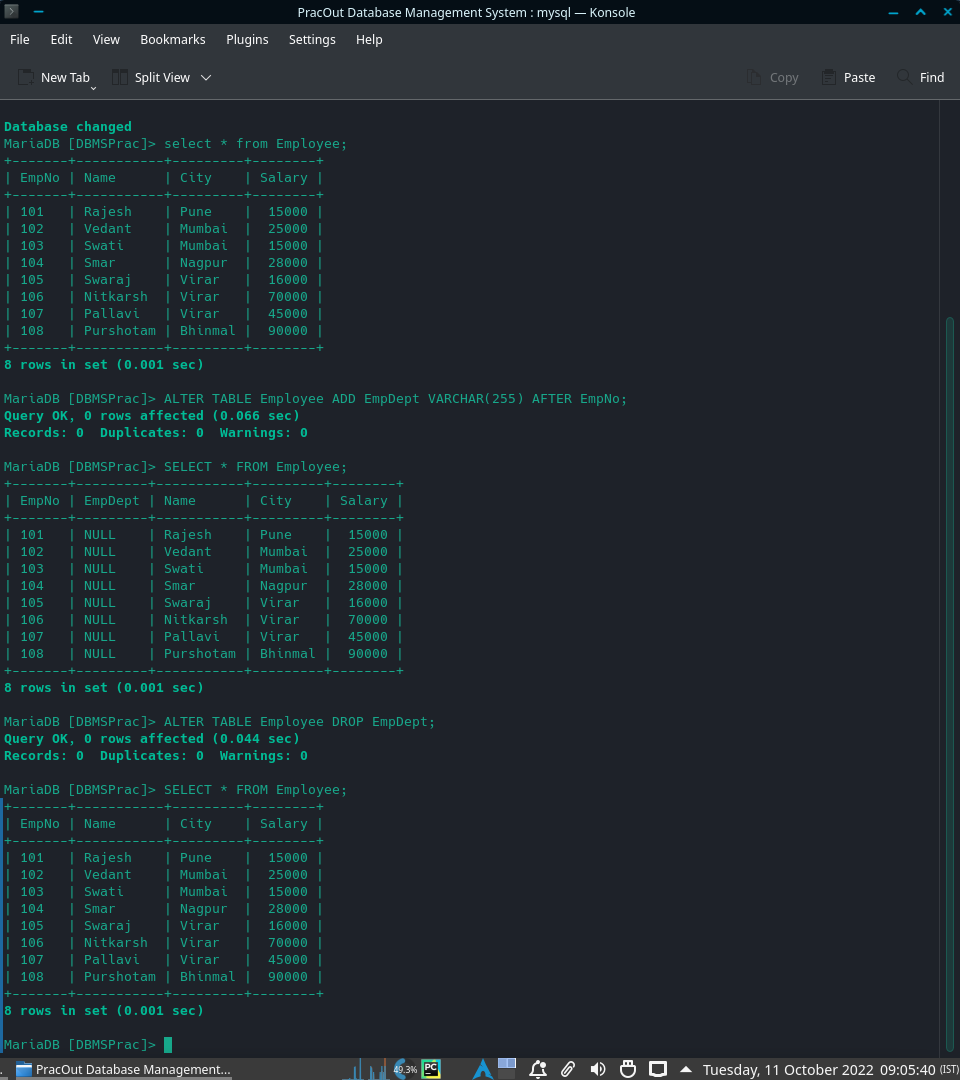
NOT NULL constraints

The NOT NULL constraint enforces a column to NOT accept NULL values

For example

Create table Persons(ID integer NOT NULL, LastName varchar(50) NOT NULL, FirstName

varchar(50) NOT NULL, Age integer(10));



Unique constraints

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

For example

Create table Persons(ID integer NOT NULL UNIQUE, LastName varchar(50) NOT NULL,

FirstName varchar(50) NOT NULL, Age integer(10));

Primary key constraints: Primary keys must contain UNIQUE values, and cannot

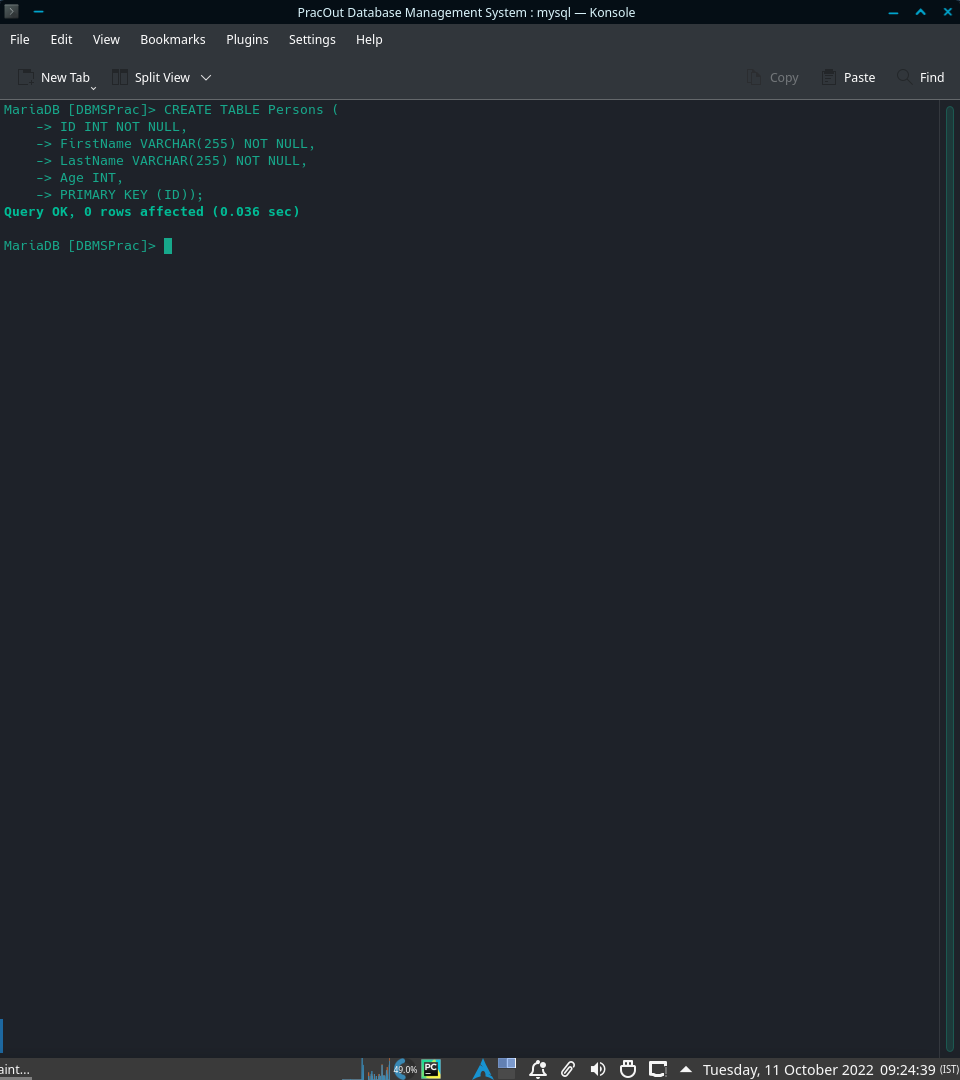
contain NULL values

Primary key on create table

For example

Create table Persons(ID integer NOT NULL UNIQUE, LastName varchar(50) NOT NULL,

FirstName varchar(50) NOT NULL, Age integer(10) PRIMARY KEY(ID));



Primary key on Alter table

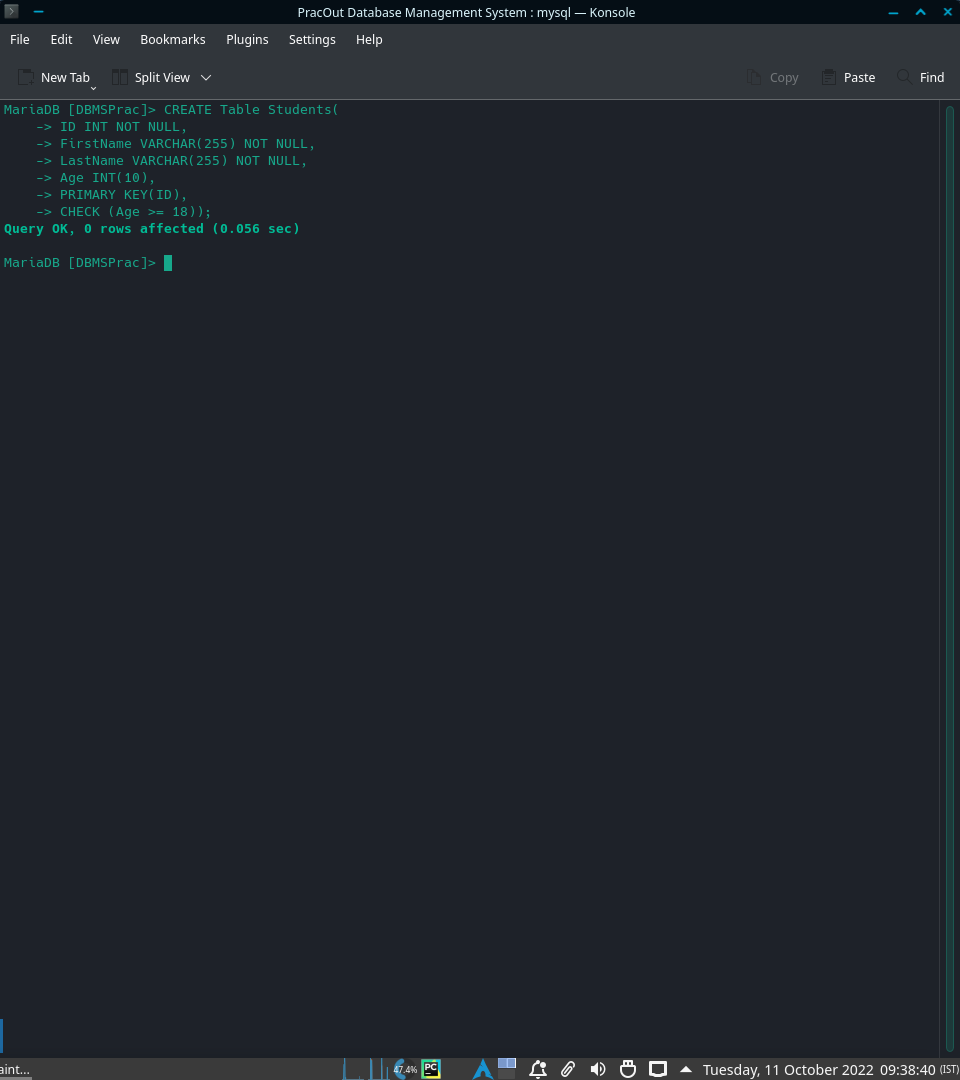
For example

[Alter table Person2 ADD PRIMARY KEY(ID); sa

Drop primary key constraints:

For example

Alter table Person2 DROP PRIMARY KEY: -



CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column

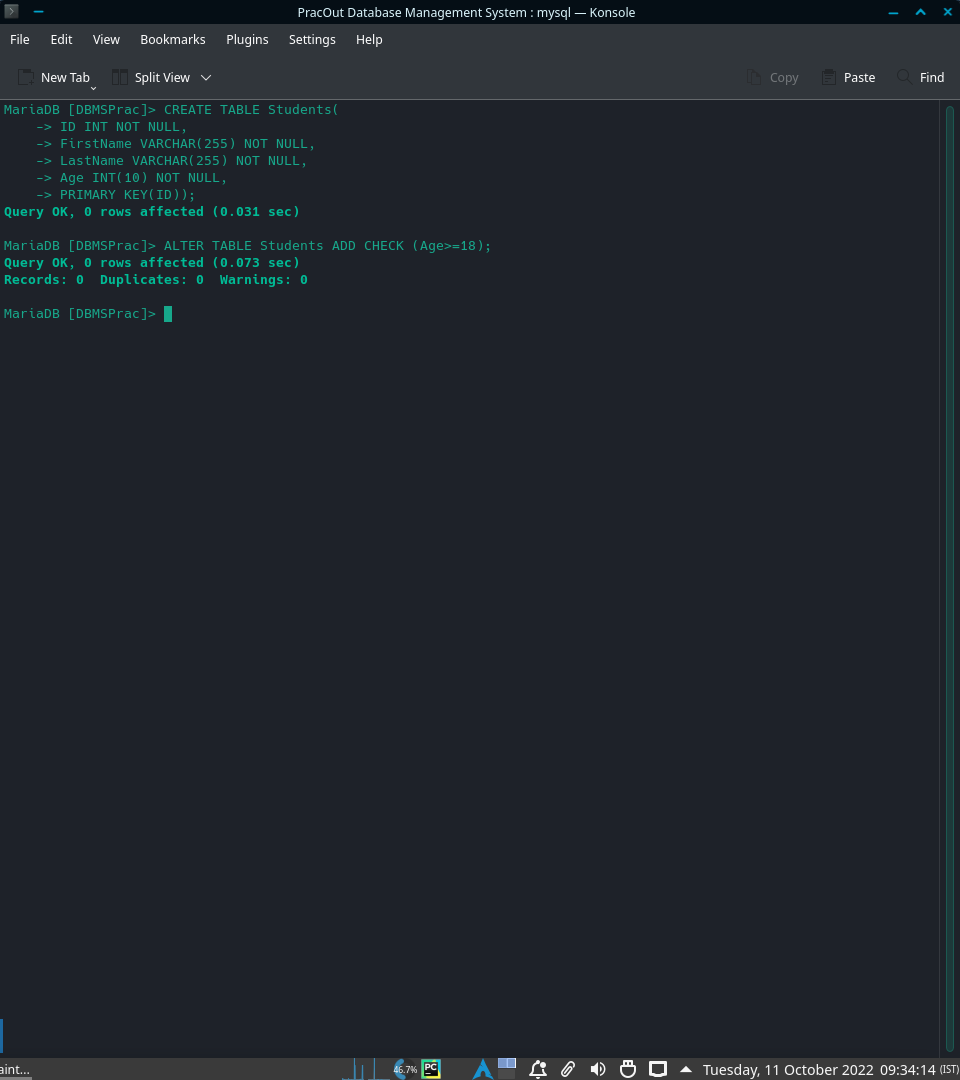
CHECK on create table

For example

create table student(ID integer NOT NULL, LastName varchar(50) NOT NULL, FirstName

| varchar(50) NOT NULL, Age integer(10) CHECK(Age>=18));

Alter table student ADD CHECK (Age>=18);



**5. Creating and Managing Other Database Objects**

(a) Creating Views

Creating View

Consider we have existing table as Employee (Employee\_no, Employee\_name,

joining\_date, Salary).

1) Creating view having all records and fields from existing table

Syntax

CREATE or replace VIEW view\_name

AS

SELECT column], column2, ...

FROM table\_name :

WHERE condition Tape or;

For Example

Create or replace view Emp\_view1

AS

select \* from Employee;

This statement will create view having all the fields as in Employee table.

2,) Creating view having specific fields but all the records from existing table

Syntax

Create or replace view view\_name

AS

select field\_1, field\_2... from existing\_table\_name;

For Example

Create or replace view Emp\_view2

As

select Employee\_no, Employee\_name from Employee;

This statement will create view having specific fields from Employee table.

3) Creating new view having specific records but all the fields from existing table

Syntax

Create or replace view view\_name

As

select \* from existing\_table\_name

where condition;

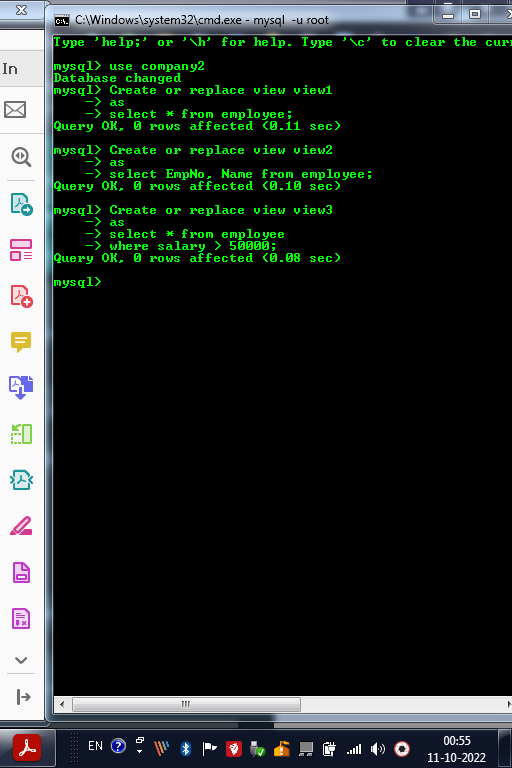
For Example

Create or replace view Emp\_view3

As

select \* from Employee

where Salary > 80000;



(b) Other Database Objects

An index is a pointer to data in a table. An index in a database is similar to the

alphabetical index of a book present at the end of book.

Indexes can be created or dropped with no effect on the data.

Creating Index

CREATE INDEX statement is used to create an index. In this statement we have to

mention name of the index, the table and column, and whether the index is in ascending or

descending order.

There are different types of indexes.

Syntax

For Example

CREATE [UNIQUE] INDEX index\_name

ON table\_name (column\_name1,[column\_name?, column\_name3,...]);

1. CREATE INDEX emp\_ind! on Employee(Employee\_name);

2.CREATE INDEX emp\_ind2 ON Employee(Employee\_no, Employee\_name),

3. CREATE UNIQUE INDEX emp\_ind3 on Employee(Employee\_name);

Displaying Index : To display index information regarding table following query is used.

Syntax

Show index from table\_name;

For Example

Show index from Employee;

Dropping Index

To drop index of a table following query is used.

Drop index emp\_ind2 on Employee;

Sequence

Sequence :

requirement.

Creating sequence

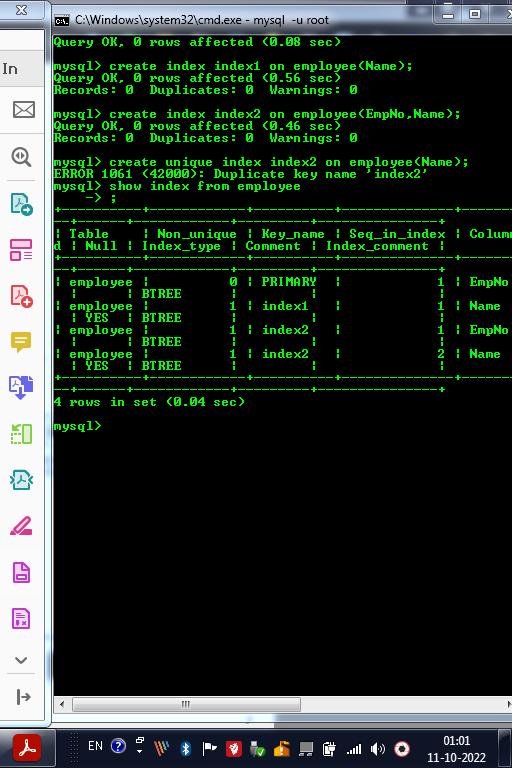
For example

create table emp2( eno integer(3) auto\_increment, primary key(eno), ename

| varchar(20),sal integer(6));

Now insert records in the table

For example



Here the column eno has auto increment values.

Synonym

Synonyms are basically alternative names for a table, view, sequence, procedure, stored

function, package etc. Synonyms provide both data independence and \_ location

transparency.

Creating Synonym in Oracle

Create synonym e for emp;

C) Controlling User Access

© Grant : to allow specified users to perform specified tasks.

o Revoke : to cancel previously granted or denied permissions.

Grant command

Syntax

GRANT <object privileges>

ON <object\_name>

TO <User\_Name>

[WITH GRANT OPTION]

Example

GRANT ALL

ON cust TO anand

WITH GRANT OPTION;

FRO IIOE LTT

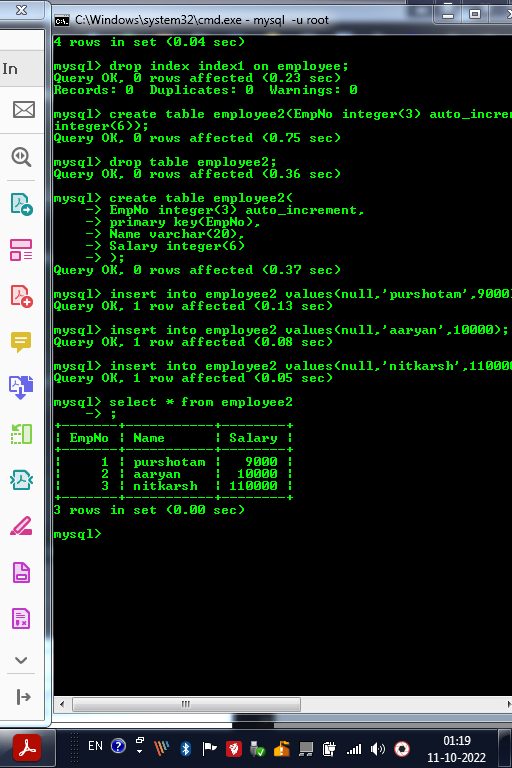
Revoke command

Syntax

REVOKE <Object\_Privileges>

ON <Object\_Name> ae

FROM <User\_Name>.



**6. Using SET operators, Date/Time Functlons, GROUP BY Seer ane**

**features) and advanced subqueries**

(a) Using SET Operators

— The different Set Operators are as follows

o Union o Union All o Intersect o Minus

Consider following two tables product\_details and Sale\_details :

Union

The union operator returns all distinct rows selected by either query

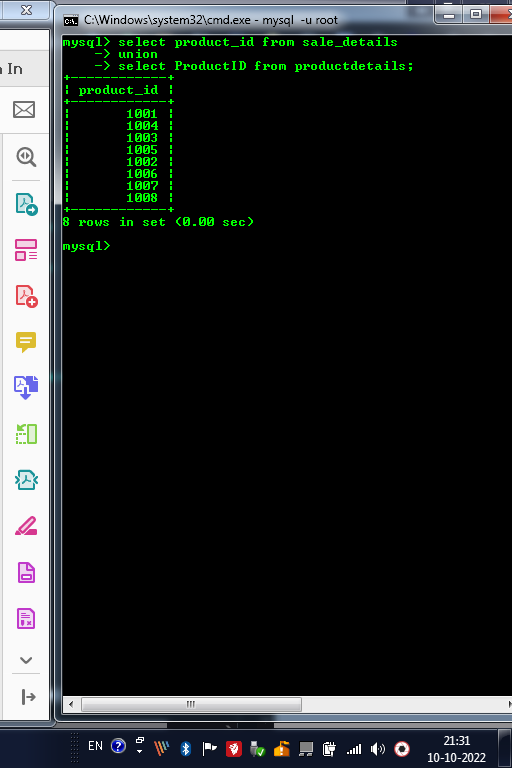
Syntax

Select column\_name from table\_1

Union

Select column\_name from table\_2 -

For Example



write a query to retrieve ids of all the products even if they were sale or present in the

storage room.

Product\_id from product\_details

Union All

The Union All operator returns all rows selected by either query including duplicates.

Syntax

| Select column\_ \_name from: tables 1

Union all

Select column. : ren able. 2

For Example

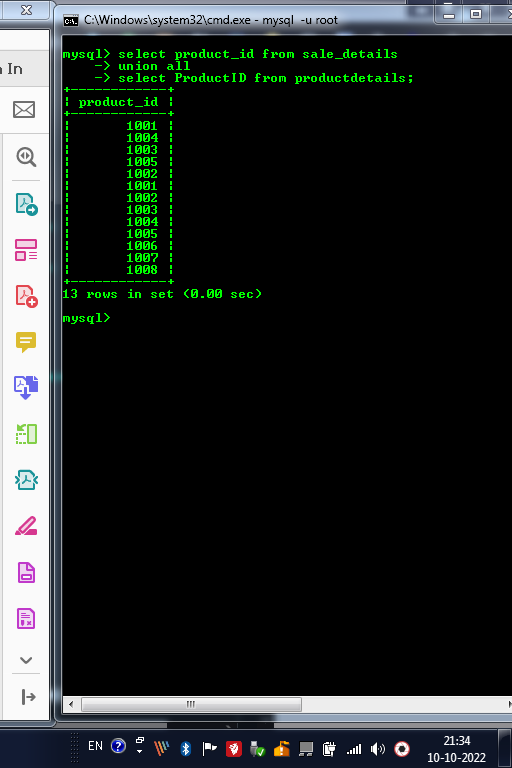
- Write a query to retrieve ids of all the products even if they were sold or present in the

Storage room.

Select Product\_id from product\_details

Union all

Select Product\_id from sale.\_etails;



Intersect

The intersect operator returns only those rows which are common to both the queries

Syntax

Select column\_name from table\_1

intersect

Select column\_name from table\_2

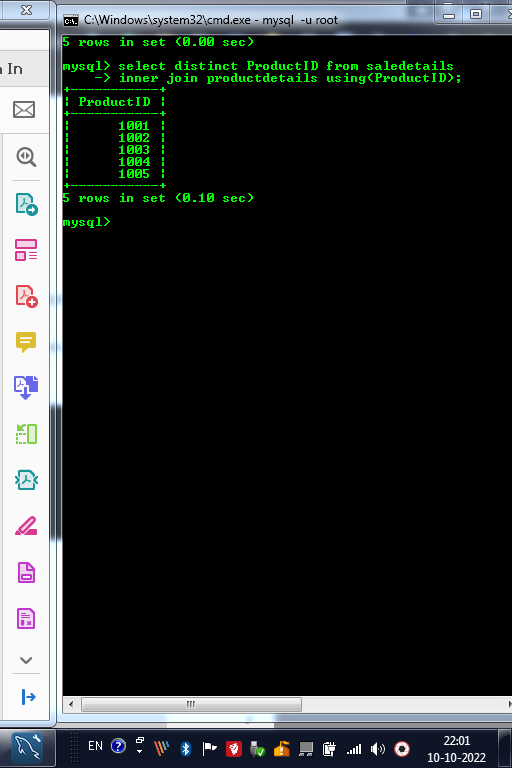
For Example

Write a query to retrieve ids of all the sold products.

Select Product\_id from proniccees

intersect

select Product\_id from sale\_ details;



Minus

Minus operator displays the rows which are present in the first query but absent in the

second query, with no duplicates and data is arranged in ascending order by default.

Syntax

Select column\_name from yes 1

minus

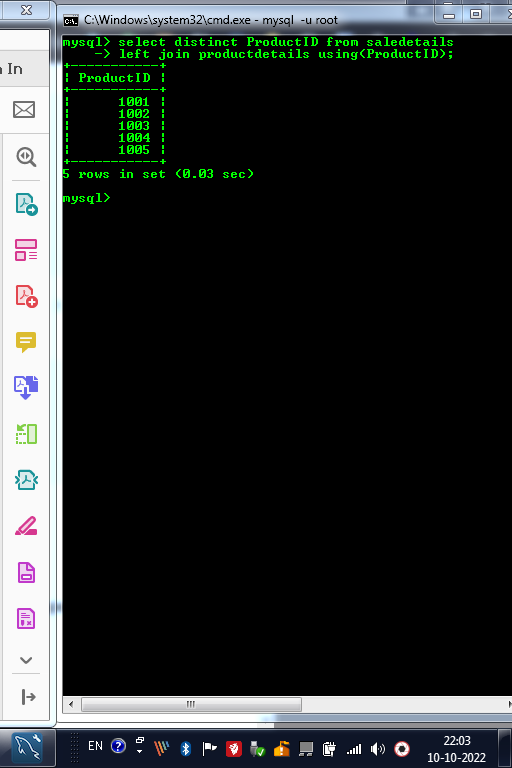
Select column\_name from aca 2%

For Example

Select Product\_id from product. details

minus

select Product\_id from sale\_details;



(b) Datetime Functions

Now0Q:

[ Now() returns the current date and time.

Example

The following SELECT statement:

SELECT NOW();

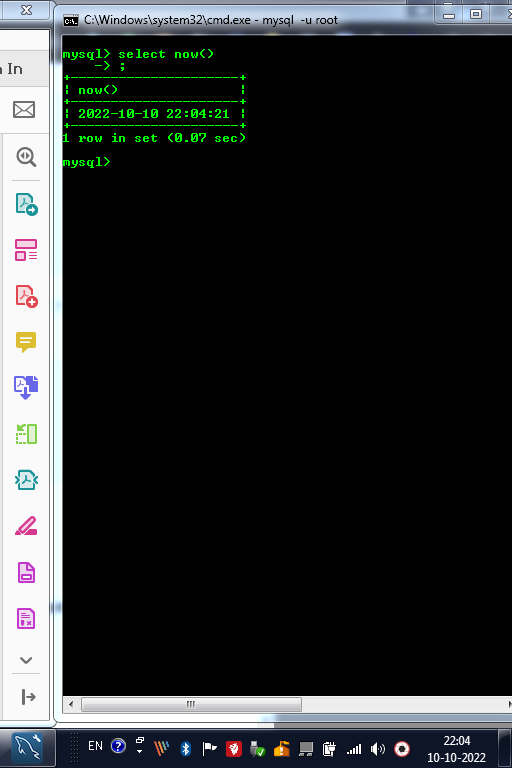
DateO

Extracts the date part of a date or date/time expression

For Example

Select date('2014-5-1));

Select date(‘2014-5-1 11:00:00.00’);



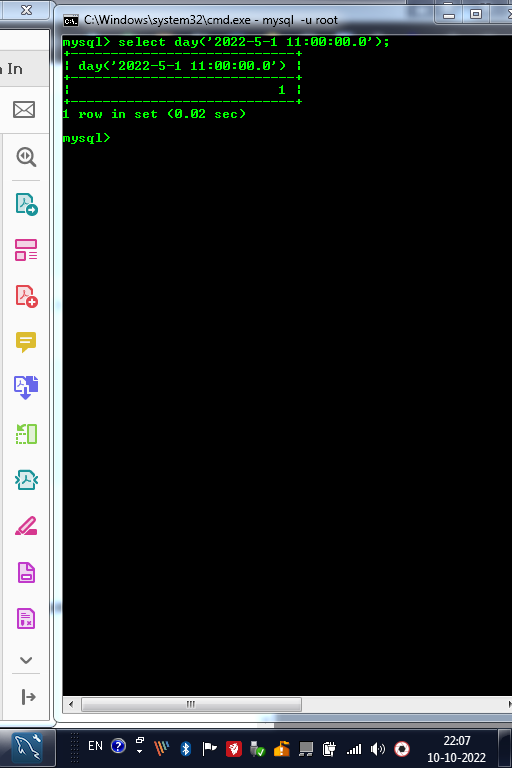
Day()

In SQL Server (Transact-SQL), the DAY function returns the day of the month (a number

from I to 31) given a date value.

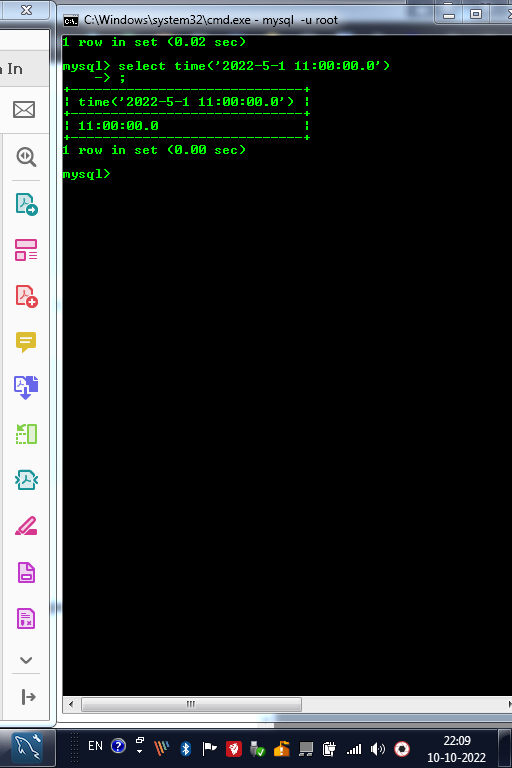
Example

Select day('2014-4-28 11:00:00.0’);



TimeQ

Select time(’11:00:00.0’);



Group by clause

GROUP BY :

The GROUP BY clause is used in collaboration with the SELBCT etatement. It helps to

arrange similar data into groups. It i¢ alsa used with SQL functions to group the femur

from one oF more tables,

Syntax

LECT column1, column2

FROM table\_name

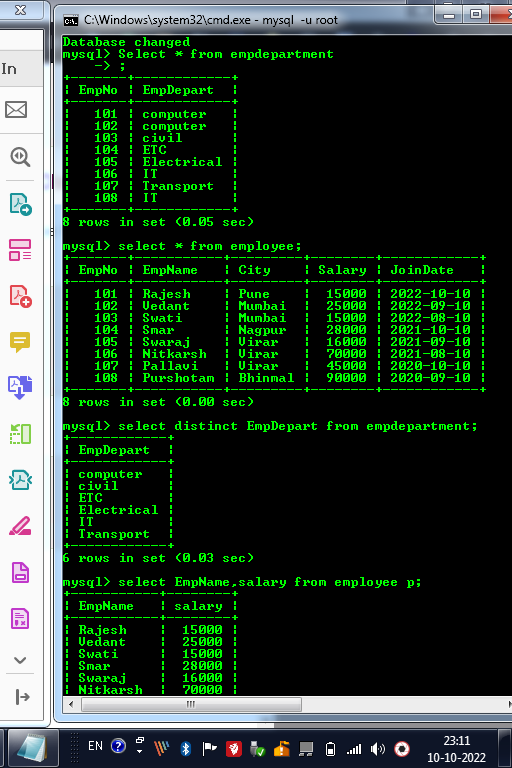
WHERE [ conditions ]

GROUP BY column1, column2

For example

Display sum of salaries department wise.

Select deptno, sum(sal) from emp group by deptno;



(d) Advanced Subqueries

In select sub query:

syntax

SELECT column\_name [, column\_name }

FROM table1 [, table2 }

WHERE column\_name OPERATOR

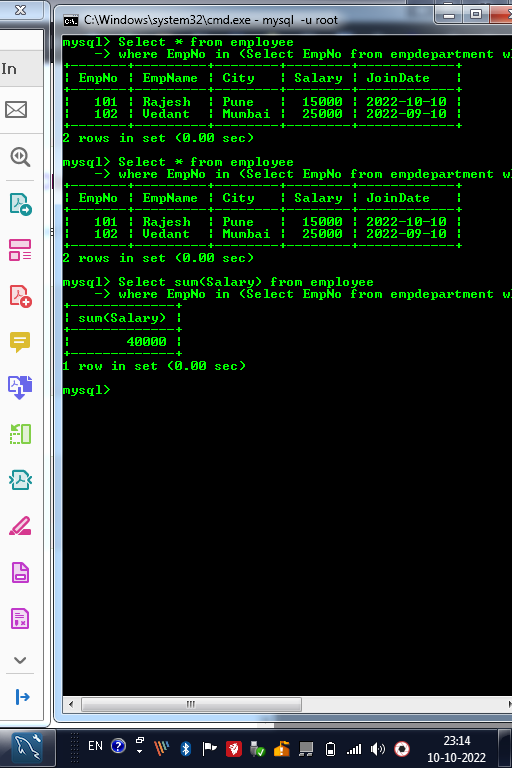
IN (SELECT column\_name [, column\_name )

FROM table1 , table2;

WHERE Employee\_no IN (SELECT Employee\_no

FROM Emp\_dept

WHERE Employee\_dept=’computer);



Update sub query

Syntax

UPDATE table

SET column\_name = new\_value

[ WHERE OPERATOR [ VALUE ] ©

(SELECT COLUMN\_NAME

FROM TABLE\_NAME) |

[WHERE)] =

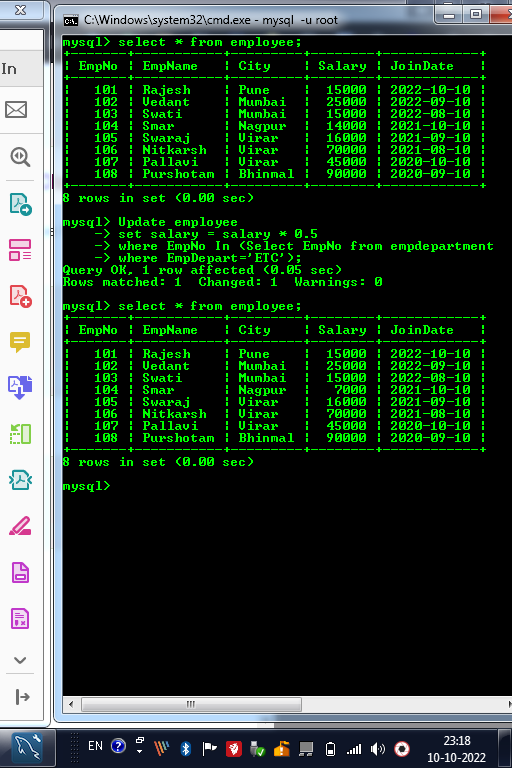
For example:

pPDATE Employee

SET SALARY = SALARY \* 0.50

WHERE Employee\_no IN (SELECT Em

WHERE Employee\_dept=’ ETC’);



Delete sub query

DELETE FROM TABLE\_NAME |

[WHERE OPERATOR [VALUE]

IN (SELECT COLUMN\_NAME

FROM TABLE\_NAME)

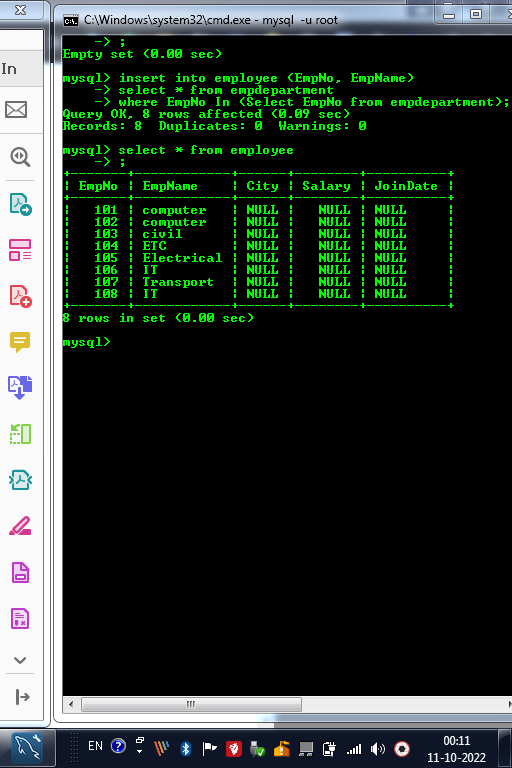
WHERE

For example

DELETE FROM Employee

WHERE Employee\_no IN ster Empaye. no FROM M Emp\_ sept be

WHERE Employee\_dept="civil’);



Insert sub query

For example

WHERE Employee\_no IN (SELECT Employee\_no FROM Emp\_dept

WHERE Employee\_ no IN (SELECT Employee ‘no , FROM Emp. dei

Update sub query

UPDATE table

seT column\_name = new value

[ WHERE OPERATOR [ VALUE ]

NOT IN (SELECT COLUMN\_NAME

FROM TABLE\_NAME)

[ WHERE) ]

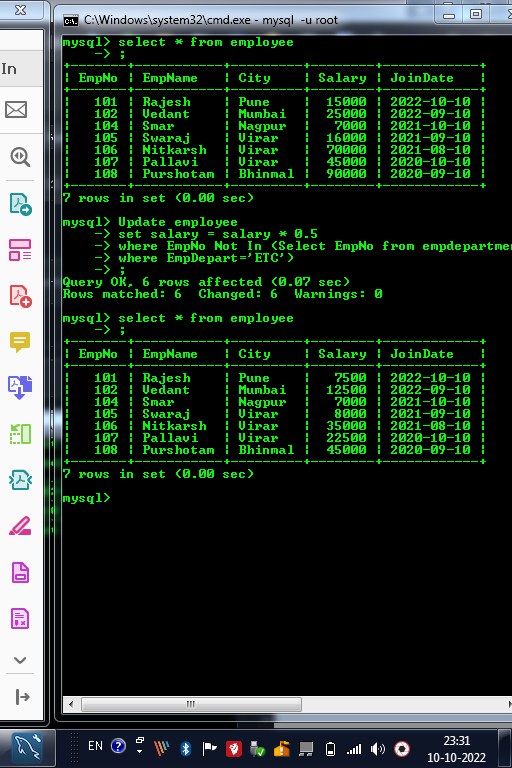
For example:

UPDATE Employee

SET SALARY = SALARY \* 0.50

WHERE Employee\_no not IN (SELECT Employee\_no FROM Emp\_dept

WHERE employes. cepts ey:



DELETE FROM TABLE\_NAME

[ WHERE OPERATOR [ VALUE ]

IN (SELECT. COLUMN NAME

FROM TABLE\_NAME)

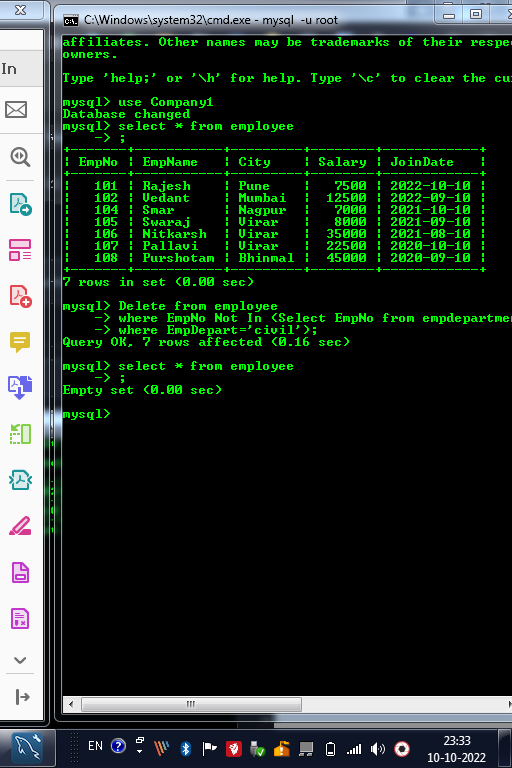
[ WHERE) ]

For example

DELETE FROM Employee

WHERE Employee\_no not IN (SELECT Employee\_no FROM Emp\_dept WHERE

Employee\_ depte'civil’);



Some

SOME compare a value to each value in a list or results from a query and evaluate to true

if the result of an inner query contains at least one row. SOME must match at least one row in

the subquery and must be preceded by comparison operators. Suppose using greater than (>)

with SOME means greater than at least one value same as it if we use less than (<) with

SOME means less than at least one value.

Syntax

SELECT {column\_name... | expression1 ]

FROM [table\_name]

WHERE expression2 comparison\_ operator {ALL 1 ANY | even ( slbqueiy )

For example

SELECT \* from Employee

WHERE no=SOME( SELECT no FROM Emp\_dept

WHERE dept='computer’);

For example

SELECT. \* from emp1

Where salary < SOME (SELECT salary FROM emp2)

WHERE city= ‘pune’);

SELECT. \* from emp2

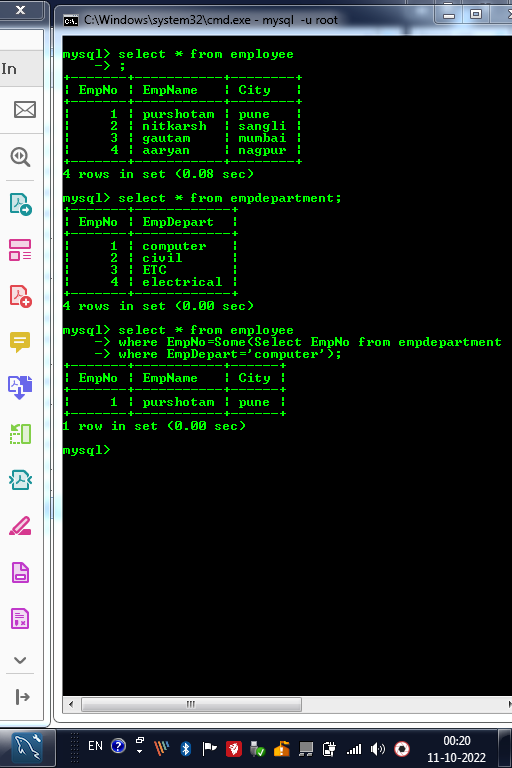
Where salary < SOME (SELECT salary FROM emp2)

WHERE city= ‘pune’);

SELECT. \* from emp3

Where salary < SOME (SELECT salary FROM emp2)

WHERE city= ‘pune’);



ALL

ALL is used to select all records of aSELECT STATEMENT. It compares a value to

every value in a list or results from a query. The ALL must be preceded by the comparison

operators and evaluates to TRUE if the query returns no rows. For example, ALL means

greater than every value, means greater than the maximum value. Suppose ALL (1, 2, 3)

means greater than 3.

Syntax

FROM [table\_name]

SELECT [column\_name...

| expression1 1 Sone

WHERE expression2 einpaneone operator {ALL ra ANY. | SOME}

**7.PL/SQL**

