**EXPERIMENT 5**

**19L-1316 Hafiz Ahmad**

# INTRODUCTION TO SQL DML

**Objective**

* Learn SQL DML commands specially the use of select query, joins and aggregate functions

## Exercise1: Practice select command

Login (172.16.1.158, sa/12345678) to **your own database** and practice the following Select commands. Observe the outputs.

1. **SELECT particular columns from a Table**

General syntax:

SELECT <list of columns> FROM <table name>

Example:

Select fname, lname, salary, bdate

From employee

1. **Select specific rows from a table**

General Syntax:

SELECT <\* or list of columns>

FROM <table name>

WHERE <condition>

Example:

Select fname, lname, salary, bdate

From employee

Where salary >25000

1. **Use of order by**

General Syntax:

SELECT <\* or list of columns>

FROM <table name>

WHERE <condition>

ORDER BY <list of columns>

Example:

Select fname, lname, salary, bdate

From employee

Where salary >25000

Order by salary desc, fname

1. **Use of DISTINCT**

Example:

Select distinct salary

From employee

1. **Use of built in functions**

Example:

Select datediff(yy, bdate,GETDATE())age,

fname from employee

1. **Use of like and between and null comparisons**

Examples:

select fname, Lname, salary, bdate

from employee

where bdate between '01-Jan-1960' and '31-Dec-1969'

and lname like 'S%'

select fname, Lname, salary, bdate

from employee

where super\_ssn is null

select fname, Lname, salary, bdate

from employee

where super\_ssn is not null

## Exercise2: Practice select command from multiple tables using inner and outer joins

1. INNER JOIN: This join only returns rows when there is at least one match in both the tables on which the JOIN is being applied.

Example:

select e.fname, e.Lname, d.Dname

from employee e innerjoin department d on e.Dno=d.Dnumber

orderby d.Dname, e.fname

1. OUTER JOIN:

There are three different Outer Join methods:

* LEFT OUTER JOIN

This join returns all the rows from the left table in conjunction with the matching rows from the right table. If there are no columns matching in the right table, it returns NULL values

Example:

select e.fname, e.Lname, d.dependent\_name

from employee e leftouterjoindependent d on e.ssn=d.essn

orderby e.fname, e.lname

* RIGHT OUTER JOIN

This join returns all the rows from the right table in conjunction with the matching rows from the left table. If there are no columns matching in the left table, it returns NULL values.

Example:

select e.fname, e.Lname, d.dependent\_name

fromdependent d right outer join employee e on e.ssn=d.essn

orderby e.fname, e.lname

* FULL OUTER JOIN

This join returns all the rows from the right as well as left table

Example:

First let us insert a department which doesn’t have a manager yet

insertinto department(Dname, dnumber)values ('Accounts', 10)

select e.fname, e.Lname, d.dName

from department d full outer join employee e on e.ssn=d.Mgr\_ssn

orderby e.fname, e.lname

## Exercise3: Practice select command using SQL Aggregate functions

1. **SQL Aggregate functions**  
   Aggregate functions perform a calculation on a set of values and return a single value. These functions are frequently used with the GROUP BY clause of the SELECT statement.

* AVG() - Returns the average of the values in a group. Null values are ignored.
* COUNT() - Returns the number of items in a group. This function always returns an int data type value
* MAX() - Returns the maximum value in the expression.
* MIN() - Returns the minimum value in the expression.
* SUM() - Returns the sum of all the values in the expression. SUM can be used on numeric columns only and it ignores all the NULL values.

General Syntax:

*SELECT <list of columns>, <aggregate functions>*

*FROM <table(s)>*

*WHERE <condition>*

*GROUP BY <column(s)>*

*HAVING < condition based on aggregate function>*

Example:

selectCOUNT(\*) NumberOfEmployee,AVG(salary) CompanyAverage,MAX(salary)CompanyMax,MIN(salary)CompanyMin,

SUM(salary) CompanySum

from employee

select d.dname,COUNT(\*) DepartmentEmps,SUM(salary) DepartmentSum

from employee e, department d

where e.Dno=d.Dnumber

group by d.dname

select d.dname,COUNT(\*) DepartmentEmps,SUM(salary) DepartmentSum

from employee e, department d

where e.Dno=d.Dnumber

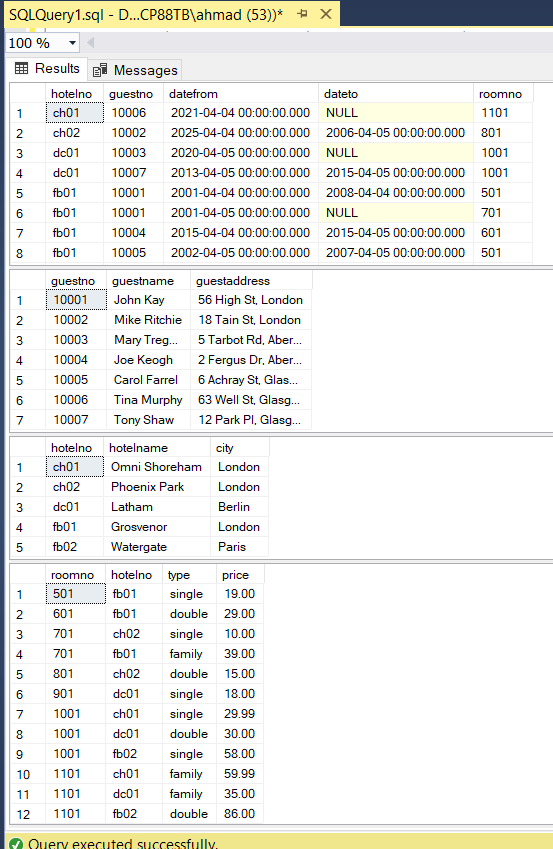
group by d.dname

having COUNT(\*)>1 and SUM(salary)>125000

Note the old syntax used for joining tables in last two queries, rewrite them using new syntax.

**EXERCISE 4:**

Using your own database, please run the lab5.sql file and write an **SQL statement along with output** to answer each of the following queries in the space provided. **Please draw the relational model for the given SQL script first.**



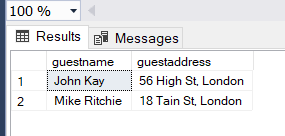
1. List the names and addresses of all guests who are from London, alphabetically ordered by guest name.

select guestname, guestaddress

from guest

where guestaddress LIKE '%London%'

order by guestname;



1. Display the names of all the guests who have not provided any end date for their reservations.

select g.guestname from guest g

join booking b on g.guestno = b.guestno

where b.dateto is null;

A screenshot of a computer

Description automatically generated

1. Display the name and city of the hotels where guests from London are staying. The list should not contain any hotel more than once.

select distinct h.hotelname, h.city

from hotel h

join room r ON h.hotelno = r.hotelno

join booking b ON r.roomno = b.roomno AND r.hotelno = b.hotelno

join guest g ON b.guestno = g.guestno

where g.guestaddress LIKE '%London%';

A screenshot of a computer

Description automatically generated

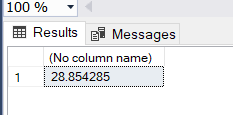
1. Display the average room price of the hotels situated in London.

SELECT AVG(r.price)

FROM hotel h

JOIN room r ON h.hotelno = r.hotelno

WHERE h.city = 'London';

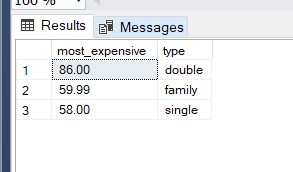


1. Display the most expensive double, single and family room respectively (across hotels).

SELECT max(r.price) most\_expensive ,r.type from hotel h

inner join room r on r.hotelno=h.hotelno

group by r.type



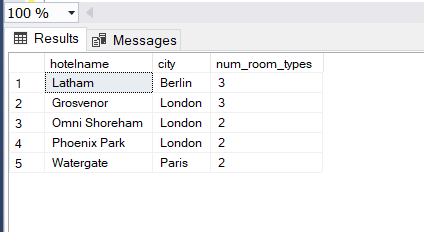
1. Display hotel name and city name along with distinct number of room types available in each of them.

SELECT h.hotelname, h.city, COUNT(DISTINCT r.type) num\_room\_types

FROM hotel h

JOIN room r ON h.hotelno = r.hotelno

GROUP BY h.hotelname, h.city;

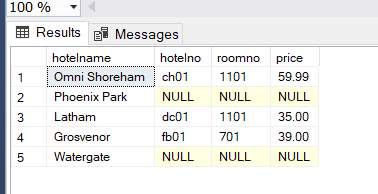


1. Display the price of the “Family” type room in all hotels along with the HotelNames, HotelNo, RoomNo. NULL should be shown if a hotel doesn’t have a “Family” type room.

SELECT h.hotelname, r.hotelno, r.roomno, r.price

FROM hotel h

LEFT JOIN room r ON h.hotelno = r.hotelno AND r.type = 'family';

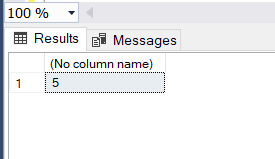


1. How many different guests have made bookings till May, 2015?

SELECT COUNT(DISTINCT b.guestno)

FROM booking b

WHERE b.datefrom <= '2015-05-31';



1. Display the name(s) of the guest(s) who have reserved two or more than two rooms in a hotel.

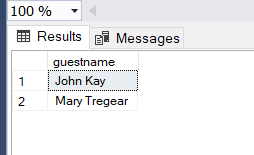
SELECT g.guestname

FROM guest AS g

JOIN booking AS b ON g.guestno = b.guestno

GROUP BY g.guestname

HAVING COUNT(b.roomno) >= 2;



1. Display the name of all the hotels in London which have more than 2 rooms.

SELECT h.hotelname

FROM hotel AS h

JOIN room AS r ON h.hotelno = r.hotelno

WHERE h.city = 'London'

GROUP BY h.hotelname

HAVING COUNT(r.roomno) > 2;

