**EXPERIMENT 6**

**Hafiz Ahmad**

# ADVANCED SQL

**Objective:**

Practice use of advanced features of the SQL select command like set functions and subqueries.

## SET FUNCTIONS

* UNION
* INTERSECT
* MINUS

## Exercise1: Practice set functions

Login to your own database and practice the following Select commands.

**Please also analyze the query and the results it is bringing and write a statement about what the query is returning on a piece of paper.**

1. select fname, lname

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

union

select e.fname, e.lname

from DEPARTMENT d innerjoin employee e on e.ssn=d.Mgr\_ssn

orderby fname, lname

--This query return two different tables and then selects the names common

-- as well as the names which are not common

1. select fname, lname

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

union all

select e.fname, e.lname

from DEPARTMENT d innerjoin employee e on e.ssn=d.Mgr\_ssn

orderby fname, lname

--This query returns two tables then prints the names common in both

-- and the names not common as well as the ones which were repeating

1. select fname, lname

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

intersect

select e.fname, e.lname

from DEPARTMENT d innerjoin employee e on e.ssn=d.Mgr\_ssn

orderby fname, lname

--selects the names which were common in both queries

1. select fname, lname

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

except

select e.fname, e.lname

from DEPARTMENT d innerjoin employee e on e.ssn=d.Mgr\_ssn

orderby fname, lname

--EXCEPT returns only rows, which are not available in the second SELECT statement.

## SUBQUERIES

There are some guidelines to consider when using subqueries:A sub query must be enclosed in parentheses.

* A sub query must be placed on the right side of the comparison operator.
* Subqueries cannot manipulate their results internally; therefore ORDER BY clause cannot be added in to a sub query.You can use an ORDER BY clause in the main SELECT statement (outer query) which will be last clause.
* Use single-row operators with single-row subqueries.
* If a sub query (inner query) returns a null value to the outer query, the outer query will not return any rows when using certain comparison operators in a WHERE clause.

**3 Subquery Types**

* Single-row subquery - where the subquery returns only one row.
* Multiple-row subquery - where the subquery returns multiple rows.
* Multiple column subquery - where the subquery returns multiple columns.

Another type of these subqueries is:

Correlated Subquery.

Are dependent on the their outer query

Will be executed many times while it’s outer queries is being processed, running once for each row selected by the outer query.

Can be in the HAVING OR WHERE clauses

## Exercise2: Practice subqueries

**Run the following queries on your own database. Please also analyze the results it is bringing and write a statement about what the query is returning on a piece of paper. Please also tell which type (single/multiple row and column, correlated or non-correlated) of subquery it is**

1. select fname, lnamefrom employee

where salary =(selectMAX(salary)from employee)

--inner query selects the maximum salary

--outer query selects the employee having the maximum salary

1. select fname, lname

from employee where ssn in(select essn fromdependent)

--inner query gives us the ssn number from the "dependent" table

--outer query selects the employees whose ssn numbers match with it

1. select fname, lname

from employee where ssn =any(select essn fromdependent)

--same query but this time we are using a unary operator "=" with a subquery

--unary operators work only when the subquery returns a single value

-- if we use "any" they can work with sub queries as well

1. select e.fname, e.Lname

from employee e whereexists(select\*fromdependent d where d.essn=e.ssn)

--same query but this time the inner query returns everything in the

--department table where the d.essn=e.ssn

-- the outer query selects the employees first and last name

1. select d.dname

from department d join dept\_locations lo on d.Dnumber=lo.Dnumber

groupby d.Dname, d.Dnumber

having COUNT(\*)>=all(selectCOUNT(\*)from dept\_locations li groupby li.dnumber)

- The query starts by selecting the `dname` (department name) from the `department` table, aliased as `d`.

- It joins the `department` table with the `dept\_locations` table, aliased as `lo`, based on the `Dnumber` column, establishing a connection between departments and their locations.

- The result is then grouped by both `dname` and `Dnumber`, creating groups for each unique department.

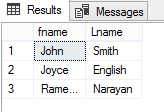
- The `HAVING` clause filters the groups, selecting only those where the count of records in `dept\_locations` for a department is greater than or equal to the count of records in `dept\_locations` for any other department.

- This way, it returns the department(s) with the highest number of locations.

**EXERCISE 3:**

Write SQL queries to answer the following questions (data from company as well as the hotel booking schema used in Lab5 will be needed). Please give your query as well as a screenshot of the output in the space provided:

1. Give the name(s) of employees who work either on project “ProductX” or on project “ProductY”

select e.fname, e.Lname

from employee as e

where e.ssn in

(

select won.essn

from works\_on as won

where won.Pno in

(

select P.Pnumber

from project P

where P.Pname='ProductX' or P.Pname='ProductY'

)

)

1. Give the name(s) of employees who work both on project “ProductX” or on project “ProductY”

select e.fname, e.Lname

from employee as e

where e.ssn in

(

select won.essn

from works\_on as won

where won.Pno in

(

select P.Pnumber

from project P

where P.Pname='ProductX'

)

)

intersect

select e.fname, e.Lname

from employee as e

where e.ssn in

(

select won.essn

from works\_on as won

where won.Pno in

(

select P.Pnumber

from project P

where P.Pname='ProductY'

)

)

1. Give the name(s) of employees who work on project “ProductY” but not on project “ProductX”

select e.fname, e.Lname

from employee as e

where e.ssn in

(

select won.essn

from works\_on as won

where won.Pno in

(

select P.Pnumber

from project P

where P.Pname='ProductY'

)

)

except

select e.fname, e.Lname

from employee as e

where e.ssn in

(

select won.essn

from works\_on as won

where won.Pno in

(

select P.Pnumber

from project P

where P.Pname='ProductX'

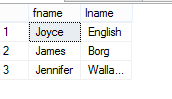
)

)

1. Give the name(s) of employees who earn the minimum salary.

select e.fname, e.lname from employee as e

where salary in (select min(e.salary) from employee as e)



1. Give the names(s) of employees who work on as many projects as “Jennifer Wallace”. Please exclude “Jennifer Wallace” from the list.

select e.fname,e.Lname

from employee as e

where e.ssn in

(

select won.essn

from works\_on as won

group by won.essn

having count(won.essn)<=all(

select count(won.essn)

from works\_on as won

where won.essn=999887777

)

)

except

select employee.fname, employee.Lname

from employee

where employee.fname='Jennifer' and employee.Lname='Wallace'

A screenshot of a computer

Description automatically generated

1. Give the names(s) of employees who work on a project which is not controlled by their own department.

select distinct e.fname, e.LnameA screenshot of a computer

Description automatically generated

from employee as e

inner join works\_on as won

on e.ssn=won.essn

inner join project as p

on won.Pno=p.Pnumber

inner join department as d

on p.Dnum=d.Dnumber

where d.Dname not in (

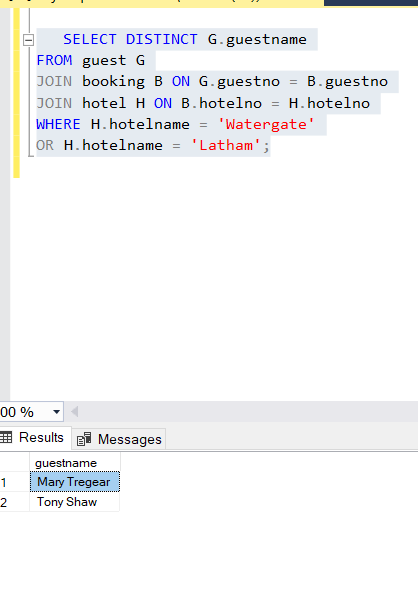
select dep.Dname

from department as dep

where e.Dno=dep.Dnumber

)

1. Give the name(s) of guest who have booking either for Watergate or Latham hotel.

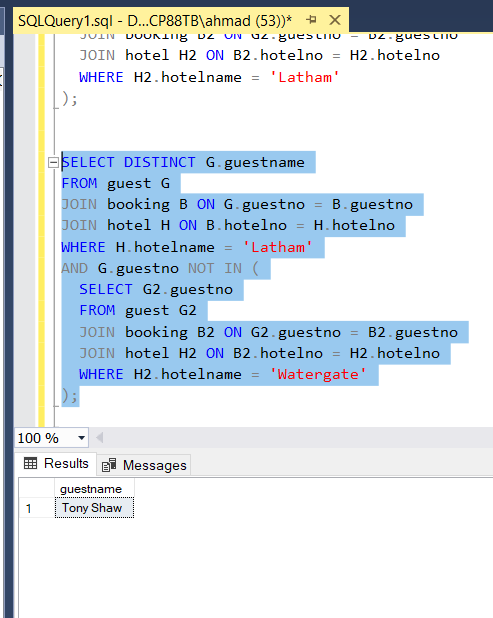


1. Give the name(s) of guest who have booking both for Watergate and Latham hotel.

A screenshot of a computer

Description automatically generated

1. Give the name(s) of guest who have booking for Latham but not for Watergate hotel.



1. Display the name of the guest who has stayed in maximum number of hotels

SELECT TOP 1 G.guestname

FROM guest G

JOIN booking B ON G.guestno = B.guestno

GROUP BY G.guestname

ORDER BY COUNT(DISTINCT B.hotelno) DESC;

A screenshot of a computer program

Description automatically generated

1. Display the name of the hotel with as many rooms as Latham hotel

SELECT H.hotelname

FROM hotel H

WHERE (

SELECT COUNT(\*)

FROM room R

WHERE R.hotelno = H.hotelno

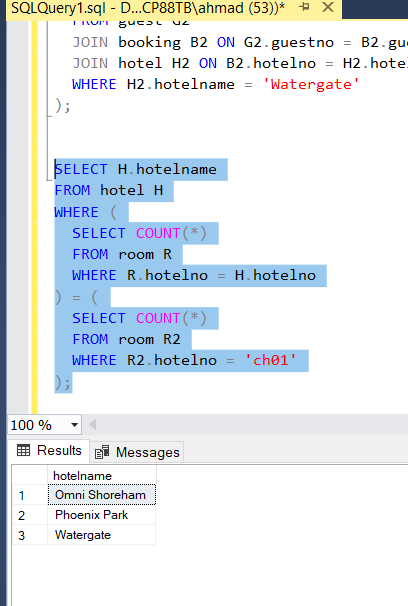
) = (

SELECT COUNT(\*)

FROM room R2

WHERE R2.hotelno = 'ch01'

)



1. Display the name of the hotel with the most expensive room

SELECT H.hotelname

FROM hotel H

WHERE (

SELECT MAX(price)

FROM room R

WHERE R.hotelno = H.hotelno

) = (

SELECT MAX(price)

FROM room R2

);

