Experiment 5: Subqueries and Views

AIM

To study and implement subqueries and views.

THEORY

Subqueries

A subquery is a query inside another SQL query and is embedded in:

- WHERE clause
- HAVING clause
- FROM clause

Types:

- *Single-row subquery*: Sub queries can also return more than one value. Such results should be made use along with the operators in and any.
- *Multiple-row subquery*: Here more than one subquery is used. These multiple sub queries are combined by means of 'and' & 'or' keywords.
- *Correlated subquery*: A subquery is evaluated once for the entire parent statement whereas a correlated Sub query is evaluated once per row processed by the parent statement.

Example: sql SELECT * FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);

Views

A view is a virtual table based on the result of an SQL SELECT query. *Create View:* sql CREATE VIEW view_name AS SELECT column1, column2 FROM table_name WHERE condition;

Drop View: sql DROP VIEW view_name;

Question 1

--

Write a SQL query that retrieve all the columns from the table "Grades", where the grade is equal to the maximum grade achieved in each subject.

Sample table: GRADES (attributes: student_id, student_name, subject, grade)

student_id	student_name	subject	grade
1	Alice	Math	90
2	Bob	Math	85
3	Charlie	Math	95
4	David	Science	88
5	Emma	Science	92
6	Frank	Science	85
7	John	Social	85

For example:

Result

student_id	student_name	subject	grade
3	Charlie	Math	95
5	Emma	Science	92
7	John	Social	85

sql

SELECT * FROM Grades g WHERE grade = (SELECT MAX(grade) FROM Grades WHERE subject = g.subject);

Output:

	Expected				Got				
✓	student_id	student_name	subject	grade	student_id	student_name	subject	grade	✓
	---	---	---	---	---	---	---	---	
	3	Charlie	Math	95	3	Charlie	Math	95	
	5	Emma	Science	92	5	Emma	Science	92	
	7	John	Social	85	7	John	Social	85	
✓	student_id	student_name	subject	grade	student_id	student_name	subject	grade	✓
	---	---	---	---	---	---	---	---	
	3	Charlie	Math	95	3	Charlie	Math	95	
	5	Emma	Science	92	5	Emma	Science	92	
	7	John	Social	85	7	John	Social	85	
	15	Sam	SOCIAL	92	15	Sam	SOCIAL	92	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

--

From the following tables write a SQL query to find all orders generated by New York-based salespeople. Return ord_no, purch_amt, ord_date, customer_id, salesman_id.

salesman table

name	type
salesman_id	numeric(5)
name	varchar(30)
city	varchar(15)
commission	decimal(5,2)

orders table

name	type
order_no	int
purch_amt	real
order_date	text
customer_id	int
salesman_id	int

For example:

Result

ord_no	purch_amt	ord_date	customer_id	salesman_id
70002	65.26	2012-10-05	3002	5001
70005	2400.6	2012-07-27	3007	5001
70008	5760.0	2012-09-10	3002	5001
70013	3045.6	2012-04-25	3002	5001

sql

```
SELECT o.ord_no, o.purch_amt, o.ord_date, o.customer_id, o.salesman_id FROM ORDERS o JOIN SALESMAN s ON o.salesman_id = s.salesman_id WHERE s.city = 'New York';
```

Output:

	Expected					Got					
✓	ord_no	purch_amt	ord_date	customer_id	salesman_id	ord_no	purch_amt	ord_date	customer_id	salesman_id	✓
	70002	65.26	2012-10-05	3002	5001	70002	65.26	2012-10-05	3002	5001	
	70005	2400.6	2012-07-27	3007	5001	70005	2400.6	2012-07-27	3007	5001	
	70008	5760.0	2012-09-10	3002	5001	70008	5760.0	2012-09-10	3002	5001	
	70013	3045.6	2012-04-25	3002	5001	70013	3045.6	2012-04-25	3002	5001	
✓	ord_no	purch_amt	ord_date	customer_id	salesman_id	ord_no	purch_amt	ord_date	customer_id	salesman_id	✓
	70002	65.26	2012-10-05	3002	5001	70002	65.26	2012-10-05	3002	5001	
	70005	2400.6	2012-07-27	3007	5001	70005	2400.6	2012-07-27	3007	5001	
	70008	5760.0	2012-09-10	3002	5001	70008	5760.0	2012-09-10	3002	5001	
	70013	3045.6	2012-04-25	3002	5001	70013	3045.6	2012-04-25	3002	5001	

Passed all tests! ✓

Correct

Question 3

--

Write a SQL query that retrieves the names of students and their corresponding grades, where the grade is equal to the maximum grade achieved in each subject.

Sample table: GRADES (attributes: student_id, student_name, subject, grade)

student_id	student_name	subject	grade
1	Alice	Math	90
2	Bob	Math	85
3	Charlie	Math	95
4	David	Science	88
5	Emma	Science	92
6	Frank	Science	85
7	John	Social	85

For example:

Result

student_name	grade
Charlie	95
Emma	92
John	85

sql

SELECT g.student_name, g.grade FROM GRADES g WHERE g.grade = (SELECT MAX(g2.grade) FROM GRADES g2 WHERE g2.subject = g.subject);

Output:

	Expected	Got	
✓	student_name grade --- --- Charlie 95 Emma 92 John 85	student_name grade --- --- Charlie 95 Emma 92 John 85	✓
✓	student_name grade --- --- Charlie 95 Emma 92 John 85 Sam 92	student_name grade --- --- Charlie 95 Emma 92 John 85 Sam 92	✓

Passed all tests! ✓

Question 4

--

Write a SQL query to Retrieve the names of customers who have a phone number that is not shared with any other customer.

SAMPLE TABLE: customer

```
name            type
--------------  ---------------
id              INTEGER
name            TEXT
city            TEXT
email           TEXT
phone           INTEGER
```

For example:

Result
name --------------- Aarti Desai Vivek Sharma Nisha Patel Rajesh Singh Radha Iyer

sql

SELECT name FROM customer WHERE phone IN (SELECT phone FROM customer GROUP BY phone HAVING COUNT(*) = 1);

Output:

	Expected	Got	
✓	name --------------- Aarti Desai Vivek Sharma Nisha Patel Rajesh Singh Radha Iyer	name --------------- Aarti Desai Vivek Sharma Nisha Patel Rajesh Singh Radha Iyer	✓
✓	name --------------- Rohit Singh Preeti Patel Manoj Gupta	name --------------- Rohit Singh Preeti Patel Manoj Gupta	✓

Passed all tests! ✓

Correct

Question 5

--

Write a SQL query to retrieve all columns from the `CUSTOMERS` table for customers whose salary is greater than $1500.

Sample table: CUSTOMERS

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000
2	Khilan	25	Delhi	1500
3	Kaushik	23	Kota	2000
4	Chaitali	25	Mumbai	6500
5	Hardik	27	Bhopal	8500
6	Komal	22	Hyderabad	4500
7	Muffy	24	Indore	10000

For example:

Result

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000
3	Kaushik	23	Kota	2000
4	Chaitali	25	Mumbai	6500
5	Hardik	27	Bhopal	8500
6	Komal	22	Hyderabad	4500
7	Muffy	24	Indore	10000

sql

```
SELECT * FROM CUSTOMERS WHERE SALARY > 1500;
```

Output:

	Expected					Got					
✓	ID	NAME	AGE	ADDRESS	SALARY	ID	NAME	AGE	ADDRESS	SALARY	✓
	1	Ramesh	32	Ahmedabad	2000	1	Ramesh	32	Ahmedabad	2000	
	3	Kaushik	23	Kota	2000	3	Kaushik	23	Kota	2000	
	4	Chaitali	25	Mumbai	6500	4	Chaitali	25	Mumbai	6500	
	5	Hardik	27	Bhopal	8500	5	Hardik	27	Bhopal	8500	
	6	Komal	22	Hyderabad	4500	6	Komal	22	Hyderabad	4500	
	7	Muffy	24	Indore	10000	7	Muffy	24	Indore	10000	
✓	ID	NAME	AGE	ADDRESS	SALARY	ID	NAME	AGE	ADDRESS	SALARY	✓
	1	Ramesh	32	Ahmedabad	2000	1	Ramesh	32	Ahmedabad	2000	
	3	Kaushik	23	Kota	2000	3	Kaushik	23	Kota	2000	
	4	Chaitali	25	Mumbai	6500	4	Chaitali	25	Mumbai	6500	
	5	Hardik	27	Bhopal	8500	5	Hardik	27	Bhopal	8500	
	6	Komal	22	Hyderabad	4500	6	Komal	22	Hyderabad	4500	
	7	Muffy	24	Indore	10000	7	Muffy	24	Indore	10000	
	15	Sam	32	Ahmedabad	4500	15	Sam	32	Ahmedabad	4500	
	17	Ram	22	Chennai	40000	17	Ram	22	Chennai	40000	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6

--

From the following tables write a SQL query to find salespeople who had more than one customer. Return salesman_id and name.

salesman table

name	type
salesman_id	numeric(5)
name	varchar(30)
city	varchar(15)
commission	decimal(5,2)

customer table

name	type
customer_id	int
cust_name	text
city	text
grade	int
salesman_id	int

For example:

Result

salesman_id	name
5001	James Hoog
5002	Nail Knite

sql

```
SELECT s.salesman_id, s.name FROM salesman s JOIN customer c ON s.salesman_id = c.salesman_id GROUP BY s.salesman_id, s.name
HAVING COUNT(c.customer_id) > 1;
```

Output:

	Expected		Got		
✓	salesman_id	name	salesman_id	name	✓
	-----------	----------	-----------	----------	
	5001	James Hoog	5001	James Hoog	
	5002	Nail Knite	5002	Nail Knite	
✓	salesman_id	name	salesman_id	name	✓
	-----------	----------	-----------	----------	
	5002	Nail Knite	5002	Nail Knite	
	5006	Mc Lyon	5006	Mc Lyon	
✓	salesman_id	name	salesman_id	name	✓
	-----------	----------	-----------	----------	
	5006	Mc Lyon	5006	Mc Lyon	
	5007	Paul Adam	5007	Paul Adam	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7

--

From the following tables write a SQL query to find the order values greater than the average order value of 10th October 2012. Return ord_no, purch_amt, ord_date, customer_id, salesman_id.

Note: date should be yyyy-mm-dd format

ORDERS TABLE

name	type
ord_no	int
purch_amt	real
ord_date	text
customer_id	int
salesman_id	int

For example:

Result

ord_no	purch_amt	ord_date	customer_id	salesman_id
70005	2400.6	2012-07-27	3007	5001
70008	5760.0	2012-09-10	3002	5001
70003	2480.4	2012-10-10	3009	5003
70013	3045.6	2012-04-25	3002	5001

sql

```
SELECT ord_no, purch_amt, ord_date, customer_id, salesman_id FROM orders WHERE purch_amt > ( SELECT AVG(purch_amt) FROM orders WHERE ord_date = '2012-10-10' );
```

Output:

	Expected					Got					
✓	ord_no	purch_amt	ord_date	customer_id	salesman_id	ord_no	purch_amt	ord_date	customer_id	salesman_id	✓
	70005	2400.6	2012-07-27	3007	5001	70005	2400.6	2012-07-27	3007	5001	
	70008	5760.0	2012-09-10	3002	5001	70008	5760.0	2012-09-10	3002	5001	
	70003	2480.4	2012-10-10	3009	5003	70003	2480.4	2012-10-10	3009	5003	
	70013	3045.6	2012-04-25	3002	5001	70013	3045.6	2012-04-25	3002	5001	
✓	ord_no	purch_amt	ord_date	customer_id	salesman_id	ord_no	purch_amt	ord_date	customer_id	salesman_id	✓
	70005	2400.6	2012-07-27	3007	5001	70005	2400.6	2012-07-27	3007	5001	
	70008	5760.0	2012-09-10	3002	5001	70008	5760.0	2012-09-10	3002	5001	
	70003	2480.4	2012-10-12	3009	5003	70003	2480.4	2012-10-12	3009	5003	
	70013	3045.6	2012-04-25	3002	5001	70013	3045.6	2012-04-25	3002	5001	

Passed all tests! ✓

Correct

Question 8

Write a SQL query to Identify customers whose city is different from the city of the customer with the highest ID

SAMPLE TABLE: customer

```
name            type
--------------  --------------
id              INTEGER
name            TEXT
city            TEXT
email           TEXT
phone           INTEGER
```

For example:

Result

```
id     name            city            email           phone
-----  --------------  --------------  --------------  ----------
6      Aarti Desai     Pune            aarti@gmail.com  890123456
7      Vivek Sharma    Chandigarh      vivek@gmail.com  980154021
8      Nisha Patel     Noida           nisha@gmail.com  901234567
9      Rajesh Singh    Hyderabad       rajesh@gmail.co  917654301
```

--

sql

SELECT * FROM customer WHERE city <> (SELECT city FROM customer WHERE id = (SELECT MAX(id) FROM customer));

Output:

	Expected					Got					
✓	id	name	city	email	phone	id	name	city	email	phone	✓
	6	Aarti Desai	Pune	aarti@gmail.com	890123456	6	Aarti Desai	Pune	aarti@gmail.com	890123456	
	7	Vivek Sharma	Chandigarh	vivek@gmail.com	980154021	7	Vivek Sharma	Chandigarh	vivek@gmail.com	980154021	
	8	Nisha Patel	Noida	nisha@gmail.com	901234567	8	Nisha Patel	Noida	nisha@gmail.com	901234567	
	9	Rajesh Singh	Hyderabad	rajesh@gmail.co	917654301	9	Rajesh Singh	Hyderabad	rajesh@gmail.co	917654301	
✓	id	name	city	email	phone	id	name	city	email	phone	✓
	1	Ravi Kumar	Delhi	ravi@gmail.com	985664321	1	Ravi Kumar	Delhi	ravi@gmail.com	985664321	
	2	Neha Sharma	Mumbai	neha@gmail.com	987654321	2	Neha Sharma	Mumbai	neha@gmail.com	987654321	
	3	Rohit Singh	Bangalore	rohit@gmail.com	887694721	3	Rohit Singh	Bangalore	rohit@gmail.com	887694721	
	4	Preeti Patel	Ahmedabad	preeti@gmail.co	789012345	4	Preeti Patel	Ahmedabad	preeti@gmail.co	789012345	
	5	Manoj Gupta	Lucknow	manoj@gmail.com	988634421	5	Manoj Gupta	Lucknow	manoj@gmail.com	988634421	

Passed all tests! ✓

Question 9

--

Write a SQL query to retrieve all columns from the CUSTOMERS table for customers whose AGE is LESS than $30

Sample table: CUSTOMERS

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000
2	Khilan	25	Delhi	1500
3	Kaushik	23	Kota	2000
4	Chaitali	25	Mumbai	6500
5	Hardik	27	Bhopal	8500
6	Komal	22	Hyderabad	4500
7	Muffy	24	Indore	10000

For example:

Result

ID	NAME	AGE	ADDRESS	SALARY
2	Khilan	25	Delhi	1500
3	Kaushik	23	Kota	2000
4	Chaitali	25	Mumbai	6500
5	Hardik	27	Bhopal	8500
6	Komal	22	Hyderabad	4500
7	Muffy	24	Indore	10000

sql

SELECT * FROM CUSTOMERS WHERE AGE < 30;

Output:

	Expected					Got					
✓	ID	NAME	AGE	ADDRESS	SALARY	ID	NAME	AGE	ADDRESS	SALARY	✓
	2	Khilan	25	Delhi	1500	2	Khilan	25	Delhi	1500	
	3	Kaushik	23	Kota	2000	3	Kaushik	23	Kota	2000	
	4	Chaitali	25	Mumbai	6500	4	Chaitali	25	Mumbai	6500	
	5	Hardik	27	Bhopal	8500	5	Hardik	27	Bhopal	8500	
	6	Komal	22	Hyderabad	4500	6	Komal	22	Hyderabad	4500	
	7	Muffy	24	Indore	10000	7	Muffy	24	Indore	10000	
✓	ID	NAME	AGE	ADDRESS	SALARY	ID	NAME	AGE	ADDRESS	SALARY	✓
	2	Khilan	25	Delhi	1500	2	Khilan	25	Delhi	1500	
	3	Kaushik	23	Kota	2000	3	Kaushik	23	Kota	2000	
	4	Chaitali	25	Mumbai	6500	4	Chaitali	25	Mumbai	6500	
	5	Hardik	27	Bhopal	8500	5	Hardik	27	Bhopal	8500	
	6	Komal	22	Hyderabad	4500	6	Komal	22	Hyderabad	4500	
	7	Muffy	24	Indore	10000	7	Muffy	24	Indore	10000	
	17	Ram	22	Chennai	1400	17	Ram	22	Chennai	1400	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10

Write a SQL query to Retrieve the medications with dosages equal to the highest dosage

Table Name: Medications (attributes: medication_id, medication_name, dosage)

name	type
medication_id	INT
medication_name	VARCHAR(50)
dosage	VARCHAR(20)

For example:

Result

```
medic  medication_name  dosage
-----  ---------------  ---------------
4      Acetaminophen    600mg
```

--

sql

select medication_id, medication_name, dosage from Medications where dosage=(select max(dosage) from Medications)

Output:

	Expected			Got			
✓	medic	medication_name	dosage	medic	medication_name	dosage	✓
	-----	---------------	---------------	-----	---------------	---------------	
	3	Amoxicillin	500mg	3	Amoxicillin	500mg	
✓	medic	medication_name	dosage	medic	medication_name	dosage	✓
	-----	---------------	---------------	-----	---------------	---------------	
	4	Acetaminophen	600mg	4	Acetaminophen	600mg	
✓	medic	medication_name	dosage	medic	medication_name	dosage	✓
	-----	---------------	---------------	-----	---------------	---------------	
	1	Aspirin	700mg	1	Aspirin	700mg	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

RESULT

Thus, the SQL queries to implement subqueries and views have been executed successfully.