

Experiment 3: DML Commands

AIM

To study and implement DML (Data Manipulation Language) commands.

THEORY

1. INSERT INTO

Used to add records into a relation. These are three type of INSERT INTO queries which are as A) Inserting a single record *Syntax (Single Row)*: `sql INSERT INTO table_name (field_1, field_2, ...) VALUES (value_1, value_2, ...);`

Syntax (Multiple Rows): `sql INSERT INTO table_name (field_1, field_2, ...) VALUES (value_1, value_2, ...), (value_3, value_4, ...);`

Syntax (Insert from another table): `sql INSERT INTO table_name SELECT * FROM other_table WHERE condition;`

2. UPDATE

Used to modify records in a relation. *Syntax*: `sql UPDATE table_name SET column1 = value1, column2 = value2 WHERE condition;`

3. DELETE

Used to delete records from a relation. *Syntax (All rows)*: `sql DELETE FROM table_name;`

Syntax (Specific condition): `sql DELETE FROM table_name WHERE condition;`

4. SELECT

Used to retrieve records from a table. *Syntax*: `sql SELECT column1, column2 FROM table_name WHERE condition;`

Question 1

--

Insert all customers from `Old_customers` into `Customers`

Table attributes are `CustomerID`, `Name`, `Address`, `Email`

For example:

Test	Result			
select * from Customers;	CustomerID	Name	Address	Email
	-----	-----	-----	-----
	301	Michael Johnson	123 Elm Street	michael.j@example.com
	302	Sarah Lee	456 Oak Avenue	sarah.lee@example.com
	303	David Wilson	789 Pine Road	david.w@example.com

Answer: (penalty regime: 0 %)

--

Insert a customer with `CustomerID` `301`, `Name` `Michael Jordan`, `Address` `123 Maple St`, `City` `Chicago`, and `ZipCode` `60616` into the `Customers` table.

For example:

Test	Result				
SELECT * FROM Customers WHERE CustomerID = 301;	CustomerID	Name	Address	City	ZipCode
	-----	-----	-----	-----	-----
	301	Michael Jordan	123 Maple St	Chicago	60616

Answer: (penalty regime: 0 %)

sql

INSERT INTO Customers(CustomerID ,Name, Address, Email) SELECT CustomerID ,Name ,Address ,Email FROM Old_customers;

sql

INSERT INTO Customers(CustomerID ,Name,Address, City , ZipCode)VALUES (301, "Michael Jordan", "123 Maple St", "Chicago", 60616);

Output:

	Test	Expected				Got				
✓	select * from Customers;	CustomerID	Name	Address	Email	CustomerID	Name	Address	Email	✓
		-----	-----	-----	-----	-----	-----	-----	-----	
		301	Michael Johnson	123 Elm Street	michael.j@example.com	301	Michael Johnson	123 Elm Street	michael.j@example.com	
		302	Sarah Lee	456 Oak Avenue	sarah.lee@example.com	302	Sarah Lee	456 Oak Avenue	sarah.lee@example.com	
		303	David Wilson	789 Pine Road	david.w@example.com	303	David Wilson	789 Pine Road	david.w@example.com	

Passed all tests! ✓

	Test	Expected					Got					
✓	SELECT * FROM Customers WHERE CustomerID = 301;	CustomerID	Name	Address	City	ZipCode	CustomerID	Name	Address	City	ZipCode	✓
		-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	
		301	Michael Jordan	123 Maple St	Chicago	60616	301	Michael Jordan	123 Maple St	Chicago	60616	

Passed all tests! ✓

Question 2

--

Write a SQL statement to update the **product_name** as 'Grapefruit' whose **product_id** is 4 in the **products** table.

products table

```
-----
product_id
product_name
category_id
availability
```

sql

UPDATE products SET product_name = 'Grapefruit' WHERE product_id =4;

Output:

	Test	Expected				Got				
✓	SELECT * FROM products WHERE product_id = 4;	product_id	product_name	category_id	availability	product_id	product_name	category_id	availability	✓
		-----	-----	-----	-----	-----	-----	-----	-----	
		4	Grapefruit	50	50	4	Grapefruit	50	50	

Passed all tests! ✓

Question 3

--

Write a SQL query to Delete a Specific Surgery whose ID is 3

Sample table: Surgeries

attributes: surgery_id, patient_id, surgeon_id, surgery_date

For example:

Test	Result			
SELECT * FROM surgeries;	surgery_id	patient_id	surgeon_id	surgery_date
	-----	-----	-----	-----
	1	1	1	2024-01-15
	2	2	2	2024-02-28
	3	3	3	2024-03-25
	surgery_id	patient_id	surgeon_id	surgery_date
	-----	-----	-----	-----
	1	1	1	2024-01-15
	2	2	2	2024-02-28

sql

DELETE FROM surgeries WHERE surgery_id =3;

Output:

	Test	Expected				Got				
✓	SELECT * FROM surgeries;	surgery_id	patient_id	surgeon_id	surgery_date	surgery_id	patient_id	surgeon_id	surgery_date	✓
		-----	-----	-----	-----	-----	-----	-----	-----	
		1	1	1	2024-01-15	1	1	1	2024-01-15	
		2	2	2	2024-02-28	2	2	2	2024-02-28	
		3	3	3	2024-03-25	3	3	3	2024-03-25	
		surgery_id	patient_id	surgeon_id	surgery_date	surgery_id	patient_id	surgeon_id	surgery_date	
		-----	-----	-----	-----	-----	-----	-----	-----	
		1	1	1	2024-01-15	1	1	1	2024-01-15	
		2	2	2	2024-02-28	2	2	2	2024-02-28	

Passed all tests! ✓

Question 4

--

Write a SQL query to locate the details of customers with grade values above 100. Return customer_id, cust_name, city, grade, and salesman_id.

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002

For example:

Result				
customer_id	cust_name	city	grade	salesman_id
3007	Brad Davis	New York	200	5001
3008	Julian Green	London	300	5002

SELECT customer_id, cust_name, city, grade, salesman_id FROM customer WHERE grade > 100;

Output:

	Expected					Got					
✓	customer_id	cust_name	city	grade	salesman_id	customer_id	cust_name	city	grade	salesman_id	✓
	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	
	3005	Graham Zusi	California	200	5002	3005	Graham Zusi	California	200	5002	
	3004	Fabian Johns	Paris	300	5006	3004	Fabian Johns	Paris	300	5006	
	3007	Brad Davis	New York	200	5001	3007	Brad Davis	New York	200	5001	
	3008	Julian Green	London	300	5002	3008	Julian Green	London	300	5002	
	3003	Jozy Altidore	Moscow	200	5007	3003	Jozy Altidore	Moscow	200	5007	
✓	customer_id	cust_name	city	grade	salesman_id	customer_id	cust_name	city	grade	salesman_id	✓
	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	
	3007	Brad Davis	New York	200	5001	3007	Brad Davis	New York	200	5001	
	3008	Julian Green	London	300	5002	3008	Julian Green	London	300	5002	

Passed all tests! ✓

Question 5

Write a SQL query to find the exact date that is 100 days after each employee's hire date.

emp table

cid	name	type
0	empno	INT
1	ename	VARCHAR(100)
2	job	VARCHAR(50)
3	mgr	INT
4	hiredate	DATE
5	sal	DECIMAL(10,2)
6	comm	DECIMAL(10,2)
7	deptno	INT

For example:

Result		
ename	hiredate	DateAfter100Days
JONES	1981-04-02	1981-07-11
MARTIN	1981-09-28	1982-01-06
BLAKE	1981-05-01	1981-08-09
CLARK	1981-06-09	1981-09-17
SCOTT	1982-12-09	1983-03-19
KING	1981-11-17	1982-02-25
TURNER	1981-09-08	1981-12-17


```
SELECT ename,hiredate,DATE(hiredate ,'+ 100 Days') AS DateAfter100Days FROM emp;
```

✓	Expected			Got			✓
	ename	hiredate	DateAfter100Days	ename	hiredate	DateAfter100Days	
✓	-----	-----	-----	-----	-----	-----	✓
	JONES	1981-04-02	1981-07-11	JONES	1981-04-02	1981-07-11	
	MARTIN	1981-09-28	1982-01-06	MARTIN	1981-09-28	1982-01-06	
	BLAKE	1981-05-01	1981-08-09	BLAKE	1981-05-01	1981-08-09	
	CLARK	1981-06-09	1981-09-17	CLARK	1981-06-09	1981-09-17	
	SCOTT	1982-12-09	1983-03-19	SCOTT	1982-12-09	1983-03-19	
	KING	1981-11-17	1982-02-25	KING	1981-11-17	1982-02-25	
	TURNER	1981-09-08	1981-12-17	TURNER	1981-09-08	1981-12-17	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Output:

Question 6

Write a SQL query to round the `decimal` column to 3 decimal places from the `Calculations` table.

cid	name	type	notnull	dflt_value	pk
0	id	INTEGER	0		1
1	value1	REAL	0		0
2	value2	REAL	0		0
3	base	INTEGER	0		0
4	exponent	INTEGER	0		0
5	number	REAL	0		0
6	decimal	REAL	0		0

For example:

Result	
id	rounded_value
1	123.457
2	567.891
3	78.234
4	45.78

sql

```
SELECT id,ROUND(decimal,3) AS rounded_value FROM Calculations;
```

Output:

	Expected		Got		
✓	id	rounded_value	id	rounded_value	✓
	-----	-----	-----	-----	
	1	123.457	1	123.457	
	2	567.891	2	567.891	
	3	78.234	3	78.234	
	4	45.78	4	45.78	

Passed all tests! ✓

Question 7

Write a SQL query to Delete customers from 'customer' table where 'CUST_NAME' has exactly 6 characters.

Sample table: Customer

CUST_CODE	CUST_NAME	CUST_CITY	WORKING_AREA	CUST_COUNTRY	GRADE	OPENING_AMT	RECEIVE_AMT	PAYMENT_AMT	OUTSTANDING_AMT	PHONE_NO	AGENT_CODE
C00013	Holmes	London	London	UK	2	6000.00	5000.00	7000.00	4000.00	BBBBBBB	A003
C00001	Micheal	New York	New York	USA	2	3000.00	5000.00	2000.00	6000.00	CCCCCCC	A008
C00020	Albert	New York	New York	USA	3	5000.00	7000.00	6000.00	6000.00	BBBBSBB	A008

For example:

Test	Result											
select changes();	CUST_CODE	CUST_NAME	CUST_CITY	WORKING_AREA	CUST_COUNTRY	GRADE	OPENING_AMT	RECEIVE_AMT	PAYMENT_AMT	OUTSTANDING_AMT	PHONE_NO	AGENT_CODE
	C00013	Holmes	London	London	UK	2	6000	5000	7000	4000	BBBBBBB	A003
	C00020	Albert	New York	New York	USA	3	5000	7000	6000	6000	BBBBSBB	A008
	C00015	Stuart	London	London	UK	1	6000	8000	3000	11000	GFSGERS	A003
	C00012	Steven	San Jose	San Jose	USA	1	5000	7000	9000	3000	KRFYGJK	A012
	C00003	Martin	Toronto	Toronto	Canada	2	8000	7000	7000	8000	MJYURFD	A004
	C00009	Ramesh	Mumbai	Mumbai	India	3	8000	7000	3000	12000	Phone No	A002
	changes()											

	6											

sql

DELETE FROM Customer WHERE LENGTH(CUST_NAME) =6;

Output:

Test	Expected													Got													
✓	select changes();	CUST_CODE	CUST_NAME	CUST_CITY	WORKING_AREA	CUST_COUNTRY	GRADE	OPENING_AMT	RECEIVE_AMT	PAYMENT_AMT	OUTSTANDING_AMT	PHONE_NO	AGENT_CODE	CUST_CODE	CUST_NAME	CUST_CITY	WORKING_AREA	CUST_COUNTRY	GRADE	OPENING_AMT	RECEIVE_AMT	PAYMENT_AMT	OUTSTANDING_AMT	PHONE_NO	AGENT_CODE	✓	
		C00013	Holmes	London	London	UK	2	6000	5000	7000	4000	8888888	A003	C00013	Holmes	London	London	UK	2	6000	5000	7000	4000	8888888	A003		
		C00020	Albert	New York	New York	USA	3	5000	7000	6000	6000	8888500	A008	C00020	Albert	New York	New York	USA	3	5000	7000	6000	6000	8888500	A008		
		C00015	Stuart	London	London	UK	1	6000	8000	3000	11000	GFSGERS	A003	C00015	Stuart	London	London	UK	1	6000	8000	3000	11000	GFSGERS	A003		
		C00012	Steven	San Jose	San Jose	USA	1	5000	7000	9000	3000	KRFYGVJK	A012	C00012	Steven	San Jose	San Jose	USA	1	5000	7000	9000	3000	KRFYGVJK	A012		
		C00003	Martin	Toronto	Toronto	Canada	2	8000	7000	7000	8000	HJYURFD	A004	C00003	Martin	Toronto	Toronto	Canada	2	8000	7000	7000	8000	HJYURFD	A004		
		C00009	Ramesh	Mumbai	Mumbai	India	3	8000	7000	3000	12000	Phone No	A002	C00009	Ramesh	Mumbai	Mumbai	India	3	8000	7000	3000	12000	Phone No	A002		
	changes()													changes()													
	-----													-----													
	6													6													
✓	select changes();	CUST_CODE	CUST_NAME	CUST_CITY	WORKING_AREA	CUST_COUNTRY	GRADE	OPENING_AMT	RECEIVE_AMT	PAYMENT_AMT	OUTSTANDING_AMT	PHONE_NO	AGENT_CODE	CUST_CODE	CUST_NAME	CUST_CITY	WORKING_AREA	CUST_COUNTRY	GRADE	OPENING_AMT	RECEIVE_AMT	PAYMENT_AMT	OUTSTANDING_AMT	PHONE_NO	AGENT_CODE	✓	
		C00013	Holmes	London	London	UK	2	6000	5000	7000	4000	8888888	A003	C00013	Holmes	London	London	UK	2	6000	5000	7000	4000	8888888	A003		
		C00001	Michel	New York	New York	USA	2	3000	5000	2000	6000	CCCCCCC	A008	C00001	Michel	New York	New York	USA	2	3000	5000	2000	6000	CCCCCCC	A008		
		C00020	Albert	New York	New York	USA	3	5000	7000	6000	6000	8888500	A008	C00020	Albert	New York	New York	USA	3	5000	7000	6000	6000	8888500	A008		
		C00015	Stuart	London	London	UK	1	6000	8000	3000	11000	GFSGERS	A003	C00015	Stuart	London	London	UK	1	6000	8000	3000	11000	GFSGERS	A003		
		C00012	Steven	San Jose	San Jose	USA	1	5000	7000	9000	3000	KRFYGVJK	A012	C00012	Steven	San Jose	San Jose	USA	1	5000	7000	9000	3000	KRFYGVJK	A012		
		C00003	Martin	Toronto	Toronto	Canada	2	8000	7000	7000	8000	HJYURFD	A004	C00003	Martin	Toronto	Toronto	Canada	2	8000	7000	7000	8000	HJYURFD	A004		
		C00009	Ramesh	Mumbai	Mumbai	India	3	8000	7000	3000	12000	Phone No	A002	C00009	Ramesh	Mumbai	Mumbai	India	3	8000	7000	3000	12000	Phone No	A002		
	changes()													changes()													
	-----													-----													
	7													7													

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8

--
Write a SQL statement to change salary of employee to 8000 whose Employee ID is 105, if the existing salary is less than 5000.

Employees table

employee_id
first_name
last_name
email
phone_number
hire_date
job_id
salary
commission_pct
manager_id
department_id

For example:

Test	Result			
SELECT EMPLOYEE_ID, FIRST_NAME, SALARY, PHONE_NUMBER FROM EMPLOYEES WHERE EMPLOYEE_ID=105;	EMPLOYEE_ID	FIRST_NAME	SALARY	PHONE_NUMBER
	-----	-----	-----	-----
	105	David	8000	590.423.4569

sql

UPDATE Employees SET salary = 8000 WHERE employee_id =105 AND salary < 5000;

Output:

	Test	Expected				Got				
✓	SELECT EMPLOYEE_ID, FIRST_NAME, SALARY, PHONE_NUMBER FROM EMPLOYEES WHERE EMPLOYEE_ID=105;	EMPLOYEE_ID	FIRST_NAME	SALARY	PHONE_NUMBER	EMPLOYEE_ID	FIRST_NAME	SALARY	PHONE_NUMBER	✓
		-----	-----	-----	-----	-----	-----	-----	-----	
		105	David	8000	590.423.4569	105	David	8000	590.423.4569	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9

Write a SQL query to categorize `value1` in the `Calculations` table as 'High' if it is greater than 50, otherwise 'Low'.

cid	name	type	notnull	dflt_value	pk
0	id	INTEGER	0		1
1	value1	REAL	0		0
2	value2	REAL	0		0
3	base	INTEGER	0		0
4	exponent	INTEGER	0		0
5	number	REAL	0		0
6	decimal	REAL	0		0

For example:

Result		
id	value1	value_category
1	-87.65	Low
2	45.78	Low
3	89.99	High
4	-0.005	Low

sql

```
SELECT id, value1, CASE WHEN value1 > 50 THEN 'High' ELSE 'Low' END AS value_category FROM Calculations;
```

	Expected			Got			
✓	id	value1	value_category	id	value1	value_category	✓
	-----	-----	-----	-----	-----	-----	
	1	-87.65	Low	1	-87.65	Low	
	2	45.78	Low	2	45.78	Low	
	3	89.99	High	3	89.99	High	
	4	-0.005	Low	4	-0.005	Low	

Passed all tests! ✓

Output: **Correct**

Question 10

--
Write a SQL query to identify products where the discount amount is greater than \$50. Return product_id, original_price, discount_percentage, and discount_amount.

Sample table: products

product_id | original_price | discount_percentage

-----+-----+-----

101 | 100.00 | 0.60

102 | 150.00 | 0.40

103 | 200.00 | 0.10

For example:

Result			
product_id	original_price	discount_percentage	discount_amount
-----	-----	-----	-----
101	100	0.6	60.0
102	150	0.4	60.0

sql

```
SELECT product_id, original_price, discount_percentage, original_price * discount_percentage AS discount_amount FROM products
WHERE
original_price * discount_percentage > 50;
```

Output:

	Expected				Got				
✓	product_id	original_price	discount_percentage	discount_amount	product_id	original_price	discount_percentage	discount_amount	✓
	-----	-----	-----	-----	-----	-----	-----	-----	
	101	100	0.6	60.0	101	100	0.6	60.0	
	102	150	0.4	60.0	102	150	0.4	60.0	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

RESULT

Thus, the SQL queries to implement DML commands have been executed successfully