

Datapark (BD) Limited

SQL Fundamentals I

Chapter 1-8

Page | 1

Chapter 04: Aggregating Data using Group Functions

1. Which three statements are true regarding the WHERE and HAVING clauses in a SQL statement? (Choose three.)

- A. The HAVING clause conditions can have aggregate functions.
- B. The HAVING clause conditions can use aliases for the columns.
- C. WHERE and HAVING clauses cannot be used together in a SQL statement.
- D. The WHERE clause is used to exclude rows before the grouping of data.
- E. The HAVING clause is used to exclude one or more aggregated results after grouping data.

Answer: A, D, E

2. Which three statements are true regarding group functions? (Choose three.)

- A. They can be used on columns or expressions.
- B. They can be passed as an argument to another group function.
- C. They can be used only with a SQL statement that has the GROUP BY clause.
- D. They can be used on only one column in the SELECT clause of a SQL statement.
- E. They can be used along with the single-row function in the SELECT clause of a SQL statement.

Answer: A, B, E

3. View the Exhibit and examine the description of the ORDERS table.

Your manager asked you to get the SALES_REP_ID and the total numbers of orders placed by each of the sales representatives. Which statement would provide the desired result?

- A.

```
SELECT sales_rep_id, COUNT(order_id) total_orders
FROM orders
GROUP BY sales_rep_id;
```
- B.

```
SELECT sales_rep_id, COUNT(order_id) total_orders
FROM orders
GROUP BY sales_rep_id, total_orders;
```
- C.

```
SELECT sales_rep_id, COUNT(order_id) total_orders
FROM orders;
```
- D.

```
SELECT sales_rep_id, COUNT(order_id) total_orders
FROM orders
WHERE sales_rep_id IS NOT NULL;
```

Answer: A

4. Which two statements are true regarding the GROUP BY clause in a SQL statement? (Choose two.)

- A. You can use column alias in the GROUP BY clause.
- B. Using the WHERE clause after the GROUP BY clause excludes the rows after creating groups.
- C. The GROUP BY clause is mandatory if you are using an aggregate function in the SELECT clause.
- D. Using the WHERE clause before the GROUP BY clause excludes the rows before creating groups.
- E. If the SELECT clause has an aggregate function, then those individual columns without an aggregate function in the SELECT clause should be included in the GROUP BY clause.

Answer: D, E

5. View the Exhibit and examine the structure of the PRODUCT_INFORMATION and INVENTORIES tables.

You want to display the quantity on hand for all the products available in the PRODUCT_INFORMATION table that have the PRODUCT_STATUS as 'orderable'. QUANTITY_ON_HAND is a column in the INVENTORIES table. The following SQL statement was written to accomplish the task:

```
SELECT pi.product_id, pi.product_status, sum(i.quantity_on_hand) FROM product_information pi LEFT OUTER JOIN inventories i
ON (pi.product_id = i.product_id) WHERE (pi.product_status = 'orderable')
GROUP BY pi.product_id, pi.product_status;
```

Which statement is true regarding the execution of this SQL statement?

- A. The statement would execute and produce the desired output.
- B. The statement would not execute because the WHERE clause is used before the GROUP BY clause.
- C. The statement would not execute because prefixing table alias to column names is not allowed with the ON clause.
- D. The statement would not execute because the WHERE clause is not allowed with LEFT OUTER JOIN.

Answer: A

Datapark (BD) Limited
SQL Fundamentals I
Chapter 1-8

6. Which are attributes of single row functions? (Choose all that apply.)

- A. cannot be nested
- B. manipulate data items
- C. act on each row returned
- D. return one result per row

Answer: B, C, D

7. What is true of using group functions on columns that contain NULL values?

- A. Group functions on columns ignore NULL values.
- B. Group functions on columns returning dates include NULL values.
- C. Group functions on columns returning numbers include NULL values.
- D. Group functions on columns cannot be accurately used on columns that contain NULL values.
- E. Group functions on columns include NULL values in calculations if you use the keyword INC_NULLS.

Answer: A

8. Which two statements are true about WHERE and HAVING clauses? (Choose two.)

- A. A WHERE clause can be used to restrict both rows and groups.
- B. A WHERE clause can be used to restrict rows only.
- C. A HAVING clause can be used to restrict both rows and groups.
- D. A HAVING clause can be used to restrict groups only.

Answer: B, D

9. Which clause should you use to exclude group results?

- A. WHERE
- B. HAVING
- C. RESTRICT
- D. GROUP BY
- E. ORDER BY

Answer: B

10. You need to calculate the total of all salaries in the accounting department. Which group function should you use?

- A. MAX
- B. MIN
- C. SUM
- D. COUNT

Answer: C

11. In a SELECT statement that includes a WHERE clause, where is the GROUP BY clause placed statement?

- A. immediately after the SELECT clause
- B. after the WHERE clause
- C. before the FROM clause
- D. after the ORDER BY clause

Answer: B

Chapter 05: Displaying Data from Multiple Tables

1. View the Exhibit and examine the structure of the ORDERS and ORDER_ITEMS tables.

Evaluate the following SQL statement:

```
SELECT oi.order_id, product_id, order_date FROM order_items oi JOIN orders o USING (order_id);
```

Which statement is true regarding the execution of this SQL statement?

- A. The statement would not execute because table aliases are not allowed in the JOIN clause.
- B. The statement would not execute because the table alias prefix is not used in the USING clause.
- C. The statement would not execute because all the columns in the SELECT clause are not prefixed with table aliases.

Datapark (BD) Limited

SQL Fundamentals I

Chapter 1-8

Page | 3

D. The statement would not execute because the column part of the USING clause cannot have a qualifier in the SELECT list.

Answer: D

2. View the Exhibit and examine the description of the DEPARTMENTS and EMPLOYEES tables.

To retrieve data for all the employees for their EMPLOYEE_ID, FIRST_NAME, and DEPARTMENT NAME, the following SQL statement was written: SELECT employee_id, first_name, department_name FROM employees NATURAL JOIN departments;
The desired output is not obtained after executing the above SQL statement. What could be the reason for this?

- A. The NATURAL JOIN clause is missing the USING clause.
- B. The table prefix is missing for the column names in the SELECT clause.
- C. The DEPARTMENTS table is not used before the EMPLOYEES table in the FROM clause.
- D. The EMPLOYEES and DEPARTMENTS tables have more than one column with the same column name and data type.

Answer: D

3. View the Exhibit and examine the description of the EMPLOYEES and DEPARTMENTS tables.

You want to display the LAST_NAME for the employees, LAST_NAME for the manager of the employees, and the DEPARTMENT_NAME for the employees having 100 as MANAGER_ID. The following SQL statement was written: SELECT m.last_name "Manager", e.last_name "Employee", department_name "Department" FROM employees m JOIN employees e ON (m.employee_id = e.manager_id) WHERE e.manager_id=100 JOIN departments d ON (e.department_id = d.department_id);
Which statement is true regarding the output of this SQL statement?

- A. The statement would provide the desired results.
- B. The statement would not execute because the ON clause is written twice.
- C. The statement would not execute because the WHERE clause is wrongly placed.
- D. The statement would not execute because the self join uses the ON clause instead of the USING clause.

Answer: C

4. View the Exhibit and examine the table structure of DEPARTMENTS and LOCATIONS tables.

You want to display all the cities that have no departments and the departments that have not been allocated cities. Which type of join between DEPARTMENTS and LOCATIONS tables would produce this information as part of its output?

- A. NATURAL JOIN
- B. FULL OUTER JOIN
- C. LEFT OUTER JOIN
- D. RIGHT OUTER JOIN

Answer: B

5. Which two statements are true regarding the types of table joins available in Oracle Database 10g? (Choose two.)

- A. You can use the JOIN clause to join only two tables.
- B. You can explicitly provide the join condition with a NATURAL JOIN.
- C. You can use the USING clause to join tables on more than one column.
- D. You can use the ON clause to specify multiple conditions while joining tables.

Answer: C, D

6. To write a query that performs an outer join of tables A and B and returns all rows from B, You need to write

- A. any outer join
- B. a left outer join
- C. a cross join
- D. a right outer join
- E. an inner join

Answer: D

Datapark (BD) Limited

SQL Fundamentals I

Chapter 1-8

Page | 4

7. Which is true regarding the use of outer joins?

- A. You cannot use IN operator in a condition that involves an outer join.
- B. You use (+) on both sides of the WHERE condition to perform an outer join.
- C. You use (*) on both sides of the WHERE condition to perform an outer join.
- D. You use an outer join to see only the rows that do not meet the join condition.

Answer: A

8. In which two cases would you use an outer join? (Choose two.)

- A. The tables being joined have NOT NULL columns.
- B. The tables being joined have only matched data.
- C. The columns being joined have NULL values.
- D. The tables being joined have only unmatched data.
- E. The tables being joined have both matched and unmatched data.
- F. Only when the tables have a primary key/foreign key relationship.

Answer: C, E

9. In which case would you use a FULL OUTER JOIN?

- A. Both tables have NULL values.
- B. You want all unmatched data from one table.
- C. You want all matched data from both tables.
- D. You want all unmatched data from both tables.

Answer: D

10. In which cases would you use the USING clause? (Choose all that apply.)

- A. You want to create a nonequijoin.
- B. The tables to be joined have multiple NULL columns.
- C. The tables to be joined have columns of the same name and different data types.
- D. The tables to be joined have columns with the same name and compatible data types.

Answer: A, D

Chapter 06: Subqueries

Question: 1

View the Exhibit and examine the description of the EMPLOYEES table.

You want to display the EMPLOYEE_ID, FIRST_NAME, and DEPARTMENT_ID for all the employees who work in the same department and have the same manager as that of the employee having EMPLOYEE_ID 104. To accomplish the task, you execute the following SQL statement:

```
SELECT employee_id, first_name, department_id
FROM employees
WHERE (manager_id, department_id)=(SELECT department_id, manager_id
FROM employees
WHERE employee_id = 104) AND employee_id <> 104;
```

When you execute the statement it does not produce the desired output. What is the reason for this?

- A. The WHERE clause condition in the main query is using the = comparison operator, instead of EXISTS.
- B. The WHERE clause condition in the main query is using the = comparison operator, instead of the IN operator.
- C. The WHERE clause condition in the main query is using the = comparison operator, instead of the = ANY operator.
- D. The columns in the WHERE clause condition of the main query and the columns selected in the subquery should be in the same order.

Answer: D

Question: 2

Which two statements are true regarding operators used with subqueries? (Choose two.)

- A. The NOT IN operator is equivalent to IS NULL.
- B. The <ANY operator means less than the maximum.
- C. =ANY and =ALL operators have the same functionality.
- D. The IN operator cannot be used in single-row subqueries.
- E. The NOT operator can be used with IN, ANY and ALL operators.

Answer: B, E

Question: 3

Which two statements are true regarding multiple-row subqueries? (Choose two.)

- A. They can contain group functions.
- B. They always contain a subquery within a subquery.
- C. They use the < ALL operator to imply less than the maximum.
- D. They can be used to retrieve multiple rows from a single table only.
- E. They should not be used with the NOT IN operator in the main query if NULL is likely to be a part of the result of the subquery.

Answer: A, E

Question: 4

View the Exhibit and examine the structure of the EMPLOYEES table
You want to know the FIRST_NAME and SALARY for all employees who have the same manager as that of the employee with the first name 'Neena' and have salary equal to or greater than that of 'Neena'.
Which SQL statement would give you the desired result?

- A. SELECT first_name, salary
FROM employees
WHERE (manager_id, salary) >= ALL (SELECT manager_id, salary
FROM employees
WHERE first_name = 'Neena') AND first_name <> 'Neena';
- B. SELECT first_name, salary
FROM employees
WHERE (manager_id, salary) >= (SELECT manager_id, salary
FROM employees
WHERE first_name = 'Neena') AND first_name <> 'Neena';
- C. SELECT first_name, salary
FROM employees
WHERE (manager_id, salary) >= ANY (SELECT manager_id, salary
FROM employees
WHERE first_name = 'Neena') AND first_name <> 'Neena';
- D. SELECT first_name, salary
FROM employees
WHERE (manager_id = (SELECT manager_id
FROM employees
WHERE first_name = 'Neena') AND salary >= (SELECT salary FROM employees
WHERE first_name = 'Neena')) AND first_name <> 'Neena';

Answer: D

Question: 5

View the Exhibit and examine the structure of the ORDERS table.

You have to display ORDER_ID, ORDER_DATE, and CUSTOMER_ID for all those orders that were placed after the last order placed by the customer whose CUSTOMER_ID is 101.
Which query would give you the desired output?

- A. SELECT order_id, order_date FROM orders
WHERE order_date > ALL (SELECT MAX(order_date) FROM orders) AND
customer_id = 101;
- B. SELECT order_id, order_date FROM orders WHERE order_date > ANY (SELECT order_date FROM orders
WHERE customer_id = 101);
- C. SELECT order_id, order_date FROM orders WHERE order_date > ALL (SELECT order_date FROM orders
WHERE customer_id = 101);
- D. SELECT order_id, order_date FROM orders
WHERE order_date IN (SELECT order_date
FROM orders
WHERE customer_id = 101);

Answer: C

Question: 6

View the Exhibit and examine the data in the PRODUCT_INFORMATION table.

Datapark (BD) Limited
SQL Fundamentals I
Chapter 1-8

PDT_ID	SUP_ID	PDT_STATUS	LIST_PRICE	MIN_PRICE
1797	102094	orderable	349	288
2254	102071	obsolete	453	371
2382	102050	under development	850	731
2459	102099	under development	699	568
3127	102087	orderable	498	444
3353	102071	obsolete	489	413
3354	102066	orderable	543	478

Which two tasks would require subqueries? (Choose two.)

- A. displaying the minimum list price for each product status
- B. displaying all supplier IDs whose average list price is more than 500
- C. displaying the number of products whose list prices are more than the average list price
- D. displaying all the products whose minimum list prices are more than the average list price of products having the product status orderable
- E. displaying the total number of products supplied by supplier 102071 and having product status OBSOLETE

Answer: C, D

Question: 7

Which two statements are true regarding subqueries? (Choose two.)

- A. Only two subqueries can be placed at one level.
- B. A subquery can be used to access data from one or more tables or views.
- C. If the subquery returns 0 rows, then the value returned by the subquery expression is NULL.
- D. The columns in a subquery must always be qualified with the name or alias of the table used.
- E. A subquery in the WHERE clause of a SELECT statement can be nested up to three levels only.

Answer: B, C

Question: 8

View the Exhibit and examine the structure of the ORDER_ITEMS table.
Examine the following SQL statement: SELECT order_id, product_id, unit_price FROM order_items WHERE unit_price = (SELECT MAX(unit_price) FROM order_items GROUP BY order_id);
You want to display the PRODUCT_ID of the product that has the highest UNIT_PRICE per ORDER_ID.
What correction should be made in the above SQL statement to achieve this?

- A. Replace = with the IN operator.
- B. Replace = with the >ANY operator.
- C. Replace = with the >ALL operator.
- D. Remove the GROUP BY clause from the subquery and place it in the main query.

Answer: A

Question: 9

View the Exhibit and examine the structure of the ORDERS table.

ORDER ID	ORDER DATE	ORDER MODE	CUST ID	ORDER TOTAL	SALES REP
2355	26-jan-98	online	104	94513.5	
2356	26-jan-00	online	105	29473.8	
2359	09-jan-98	online	106	5543.1	
2360	14-nov-99	online	107	990.4	
2432	14-sep-99	direct	102	10523	163
2433	13-sep-99	direct	103	78	163
2436	02-sep-99	direct	116	6394.8	161
2446	27-jul-99	direct	117	103679.3	161
2447	27-jul-00	direct	101	33893.6	161

Which task would require subqueries?

- A. displaying the total order value for sales representatives 161 and 163

Datapark (BD) Limited

SQL Fundamentals I

Chapter 1-8

Page | 7

- B. displaying the order total for sales representative 161 in the year 1999
- C. displaying the number of orders that have order mode online and order date in 1999
- D. displaying the number of orders whose order total is more than the average order total for all online orders

Answer: D

Question: 10

A subquery is called a single-row subquery when ____.

- A. the inner query returns a single value to the main query
- B. the inner query uses an aggregate function and returns one or more values
- C. there is only one inner query in the main query and the inner query returns one or more values
- D. the inner query returns one or more values and the main query returns a single value as output

Answer: A

Chapter07: Using the Set Operators

Question: 1

View the Exhibit and examine the structure of the EMPLOYEES and DEPARTMENTS tables.

Which SET operator would you use in the blank space in the following SQL statement to list the departments where all the employees have managers?

```
SELECT department_id
FROM departments
```

```
____
SELECT department_id
FROM employees
WHERE manager_id IS NULL;
```

- A. UNION B. MINUS
- C. INTERSECT D. UNION ALL

Answer: B

Question: 2

View the Exhibit and examine the structure of the LOCATIONS and DEPARTMENTS tables.

Which SET operator should be used in the blank space in the following SQL statement to display the cities that have departments located in them?

```
SELECT location_id, city
FROM locations
```

```
SELECT location_id, city
FROM locations JOIN departments
USING(location_id);
```

- A. UNION
- B. MINUS
- C. INTERSECT
- D. UNION ALL

Answer: C

Question: 3

View the Exhibit and examine the data in the EMPLOYEES tables.

Evaluate the following SQL statement: SELECT employee_id, department_id FROM employees WHERE department_id= 50 ORDER BY department_id UNION SELECT employee_id, department_id FROM employees WHERE department_id= 90 UNION SELECT employee_id, department_id FROM employees WHERE department_id= 10;

What would be the outcome of the above SQL statement?

- A. The statement would execute successfully and display all the rows in the ascending order of DEPARTMENT_ID.
- B. The statement would execute successfully but it will ignore the ORDER BY clause and display the rows in random order.
- C. The statement would not execute because the positional notation instead of the column name should be used with the ORDER BY clause.
- D. The statement would not execute because the ORDER BY clause should appear only at the end of the SQL statement, that is, in the last SELECT statement.

Answer: D

Question: 4

View the Exhibit and examine the structure of the EMPLOYEES and JOB_HISTORY tables.

The query should display the employee IDs of all the employees who have held the job SA_MAN at any time during their tenure.
Choose the correct SET operator to fill in the blank space and complete the following query. SELECT employee_id
FROM employees
WHERE job_id = 'SA_MAN'
_____ SELECT employee_id FROM job_history
WHERE job_id='SA_MAN';

- A. UNION
- B. MINUS
- C. INTERSECT
- D. UNION ALL

Answer: A

Question: 5

Evaluate the following SQL statement: SELECT 2 col1,'y' col2
FROM dual UNION SELECT 1,'x' FROM dual UNION
SELECT 3,NULL

FROM dual
ORDER BY 2;
Which statement is true regarding the output of the SQL statement?

- A. It would execute and the order of the values in the first column would be 3, 2, 1.
- B. It would execute and the order of the values in the first column would be 1, 2, 3.
- C. It would not execute because the column alias name has not been used in the ORDER BY clause.
- D. It would not execute because the number 2 in the ORDER BY clause would conflict with the value 2 in the first SELECT statement.

Answer: B

Chapter 08: Manipulating Data

Question: 1

View the Exhibit and examine the structure of the CUST table.

CUST		
Name	Null?	Type
CUST_ID	NOT NULL	NUMBER(2)
CUST_NAME		VARCHAR2(15)

Evaluate the following SQL statements executed in the given order: ALTER TABLE cust
ADD CONSTRAINT cust_id_pk PRIMARY KEY(cust_id) DEFERRABLE INITIALLY DEFERRED; INSERT INTO cust VALUES
(1,'RAJ'); --row 1
INSERT INTO cust VALUES (1,'SAM'); --row 2
COMMIT;
SET CONSTRAINT cust_id_pk IMMEDIATE; INSERT INTO cust VALUES (1,'LATA'); --row 3
INSERT INTO cust VALUES (2,'KING'); --row 4
COMMIT;
Which rows would be made permanent in the CUST table?

- A. row 4 only

- B. rows 2 and 4
- C. rows 3 and 4
- D. rows 1 and 4

Answer: C

Question: 2

Given below are the SQL statements executed in a user session:

```
CREATE TABLE product
(pcode NUMBER(2), pname VARCHAR2(10));
INSERT INTO product VALUES(1, 'pen'); INSERT INTO product VALUES (2,'pencil'); SAVEPOINT a;
UPDATE product SET pcode = 10 WHERE pcode = 1; SAVEPOINT b;
DELETE FROM product WHERE pcode = 2; COMMIT;
DELETE FROM product WHERE pcode=10; ROLLBACK TO SAVEPOINT a;
```

Which statement describes the consequences?

- A. No SQL statement would be rolled back.
- B. Both the DELETE statements would be rolled back.
- C. Only the second DELETE statement would be rolled back.
- D. Both the DELETE statements and the UPDATE statement would be rolled back.

Answer: A

Question: 3

View the Exhibit and examine the structure of the ORDERS table.

ORDERS		
Name	Null?	Type
ORDER ID	NOT NULL	NUMBER(12)
ORDER DATE		DATE
CUSTOMER ID		NUMBER(6)
ORDER TOTAL		NUMBER(8,2)

The ORDERS table belongs to the user OE. HR is another user in the database. Evaluate the commands issued by users OE and HR in the following order: Statement 1 by user OE: GRANT SELECT, UPDATE(customer_id, order_total) ON orders TO hr; Statement 1 by user HR: SELECT * FROM oe.orders; Statement 2 by user HR: UPDATE oe.orders SET order_total= 10000; Which statement is true regarding the above commands?

- A. Statement 1 by user OE would not work because the statement has to be issued by the DBA.
- B. Statement 2 by user HR would not work because the grant is only for SELECT in a subquery of update.
- C. There are no errors in the statements issued by OE and HR; all the statements would execute successfully.
- D. Statement 1 by user HR would not work because SELECT and UPDATE privileges have been granted only on CUSTOMER_ID and ORDER_TOTAL columns.

Answer: C

Question: 4

View the Exhibit and examine the structure of the ORDERS table.

NEW_ORDERS is a new table with the columns ORD_ID, ORD_DATE, CUST_ID, and ORD_TOTAL that have the same data types and size as the corresponding columns in the ORDERS table. Evaluate the following INSERT statement: INSERT INTO new_orders (ord_id, ord_date, cust_id, ord_total) VALUES(SELECT order_id,order_date,customer_id,order_total FROM orders WHERE order_date > '31-dec-1999'); Why would the INSERT statement fail?

- A. because column names in NEW_ORDERS and ORDERS tables do not match
- B. because the VALUES clause cannot be used in an INSERT with a subquery
- C. because the WHERE clause cannot be used in a subquery embedded in an INSERT statement
- D. because the total number of columns in the NEW_ORDERS table does not match the total number of columns in the ORDERS table

Answer: B

Question: 5

View the Exhibit and examine the structure of the ORDERS table.

Which UPDATE statement is valid?

- A. UPDATE orders
SET order_date = '12-mar-2007', order_total IS NULL
WHERE order_id = 2455;
- B. UPDATE orders
SET order_date = '12-mar-2007', order_total = NULL
WHERE order_id = 2455;
- C. UPDATE orders
SET order_date = '12-mar-2007'
AND order_total = TO_NUMBER(NULL) WHERE order_id = 2455;
- D. UPDATE orders
SET order_date = TO_DATE('12-mar-2007','dd-mon-yyyy'), SET order_total = TO_NUMBER(NULL)
WHERE order_id = 2455;

Answer: A

Question: 6

Which three statements indicate the end of a transaction? (Choose three.)

- A. after a COMMIT is issued
- B. after a ROLLBACK is issued
- C. after a SAVEPOINT is issued
- D. after a SELECT statement is issued
- E. after a CREATE statement is issued

Answer: A, B, E

Question: 7

View the Exhibit and examine the descriptions of the DEPT and LOCATIONS tables.

You want to update the CITY column of the DEPT table for all the rows with the corresponding value in the CITY column of the LOCATIONS table for each department.
Which SQL statement would you execute to accomplish the task?

- A. UPDATE dept d
SET city = ANY (SELECT city
FROM locations l); B. UPDATE dept d
SET city = (SELECT city
FROM locations l)
WHERE d.location_id = l.location_id; C. UPDATE dept d
SET city = (SELECT city
FROM locations l
WHERE d.location_id = l.location_id); D. UPDATE dept d
SET city = ALL (SELECT city
FROM locations l
WHERE d.location_id = l.location_id);

Answer: C

Question: 8

Evaluate the following DELETE statement: DELETE FROM orders;
There are no other uncommitted transactions on the ORDERS table. Which statement is true about the DELETE statement?

- A. It removes all the rows in the table and allows ROLLBACK.
- B. It would not remove the rows if the table has a primary key.
- C. It removes all the rows as well as the structure of the table.
- D. It removes all the rows in the table and does not allow ROLLBACK.

Answer: A

Question: 9

View the Exhibit and examine the structure of ORDERS and ORDER_ITEMS tables.
ORDER_ID is the primary key in the ORDERS table.
It is also the foreign key in the ORDER_ITEMS table wherein it is created with the ON DELETE CASCADE option.

Datapark (BD) Limited

SQL Fundamentals I

Chapter 1-8

Which DELETE statement would execute successfully?

- A. DELETE order_id
FROM orders
WHERE order_total < 1000;
- B. DELETE orders
WHERE order_total < 1000;
- C. DELETE
FROM orders
WHERE (SELECT order_id
FROM order_items);
- D. DELETE orders o, order_items i
WHERE o.order_id = i.order_id;

Answer: B

Question: 10

View the Exhibit and examine the structure of EMPLOYEES and JOB_HISTORY tables.

The EMPLOYEES table maintains the most recent information regarding salary, department, and job for all the employees. The JOB_HISTORY table maintains the record for all the job changes for the employees. You want to delete all the records from the JOB_HISTORY table that are repeated in the EMPLOYEES table.

Which two SQL statements can you execute to accomplish the task? (Choose two.)

- A. DELETE
FROM job_history j
WHERE employee_id =
(SELECT employee_id
FROM employees e
WHERE j.employee_id = e.employee_id) AND job_id = (SELECT job_id
FROM employees e
WHERE j.job_id = e.job_id);
- B. DELETE
FROM job_history j
WHERE (employee_id, job_id) = ALL
(SELECT employee_id, job_

FROM employees e
WHERE j.employee_id = e.employee_id and j.job_id = e.job_id)
- C. DELETE
FROM job_history j
WHERE employee_id =
(SELECT employee_id
FROM employees e
WHERE j.employee_id = e.employee_id and j.job_id = e.job_id)
- D. DELETE
FROM job_history j
WHERE (employee_id, job_id) =
(SELECT employee_id, job_id
FROM employees e
WHERE j.employee_id = e.employee_id and j.job_id = e.job_id)

Answer: C, D

Question: 11

View the Exhibit and examine the structure of ORDERS and CUSTOMERS tables.

Evaluate the following UPDATE statement: UPDATE
(SELECT order_date, order_total, customer_id
FROM orders)
SET order_date = '22-mar-2007' WHERE customer_id =
(SELECT customer_id
FROM customers
WHERE cust_last_name = 'Roberts' AND
credit_limit = 600);

Which statement is true regarding the execution of the above UPDATE statement?

- A. It would not execute because two tables cannot be used in a single UPDATE statement.
- B. It would execute and restrict modifications to only the columns specified in the SELECT statement.
- C. It would not execute because a subquery cannot be used in the WHERE clause of an UPDATE statement.
- D. It would not execute because the SELECT statement cannot be used in place of the table name.

Answer: B

Question: 12

View the Exhibit and examine the structure of ORDERS and CUSTOMERS tables.

ORDERS

Name	Null?	Type
ORDER_ID	NOT NULL	NUMBER (4)
ORDER_DATE	NOT NULL	DATE
ORDER_MODE		VARCHAR2 (8)
CUSTOMER_ID	NOT NULL	NUMBER (6)
ORDER_TOTAL		NUMBER (8, 2)

CUSTOMERS

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER (6)
CUST_FIRST_NAME	NOT NULL	VARCHAR2 (20)
CUST_LAST_NAME	NOT NULL	VARCHAR2 (20)
CREDIT_LIMIT		NUMBER (9, 2)
CUST_ADDRESS		VARCHAR2 (40)

Which INSERT statement should be used to add a row into the ORDERS table for the customer whose CUST_LAST_NAME is Roberts and CREDIT_LIMIT is 600?

- A. INSERT INTO orders
VALUES (1,'10-mar-2007', 'direct',
(SELECT customer_id
FROM customers
WHERE cust_last_name='Roberts' AND
credit_limit=600), 1000);
- B. INSERT INTO orders (order_id,order_date,order_mode,
(SELECT customer_id
FROM customers
WHERE cust_last_name='Roberts' AND
credit_limit=600),order_total)
VALUES(1,'10-mar-2007', 'direct', &&customer_id, 1000);
- C. INSERT INTO orders (order_id,order_date,order_mode,
(SELECT customer_id
FROM customers
WHERE cust_last_name='Roberts' AND
credit_limit=600),order_total)
VALUES(1,'10-mar-2007', 'direct', &customer_id, 1000);
- D. INSERT INTO(SELECT o.order_id, o.order_date,o.order_mode,c.customer_id, o.order_total
FROM orders o, customers c
WHERE o.customer_id = c.customer_id
AND c.cust_last_name='Roberts' ANDc.credit_limit=600) VALUES (1,'10-mar-2007', 'direct',(SELECT customer_id FROM
customers
WHERE cust_last_name='Roberts' AND
credit_limit=600), 1000);

Answer: A

Question: 13

View the Exhibit and examine the data in EMPLOYEES and DEPARTMENTS tables.

Datapark (BD) Limited
SQL Fundamentals I
Chapter 1-8

EMPLOYEES					
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID	SALARY	COMMISSION_PCT
154	Nanette	Cambrault	80	7500	.2
166	Sundar	Aude	80	6400	.1
167	Amit	Banda	80	6200	.1
169	Harrison	Bloom	80	10000	.2

DEPARTMENTS		
DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
40	Human Resources	2400
70	Public Relations	2700
80	Sales	2500

In the EMPLOYEES table EMPLOYEE_ID is the PRIMARY KEY and DEPARTMENT_ID is the FOREIGN KEY. In the DEPARTMENTS table DEPARTMENT_ID is the PRIMARY KEY. Evaluate the following UPDATE statement:

```
UPDATE employees a
SET department_id =
(SELECT department_id
FROM departments
WHERE location_id = '2100'),
(salary, commission_pct) =
(SELECT 1.1*AVG(salary), 1.5*AVG(commission_pct) FROM employees b
WHERE a.department_id = b.department_id) WHERE first_name||' '||last_name = 'Amit Banda';
```

What would be the outcome of the above statement?

A. It would execute successfully and update the relevant data.
B. It would not execute successfully because there is no LOCATION_ID 2100 in the DEPARTMENTS table.
C. It would not execute successfully because the condition specified with the concatenation operator is not valid.
D. It would not execute successfully because multiple columns (SALARY,COMMISSION_PCT)cannot be used in an UPDATE statement.

Answer: A

Question: 14

You executed the following SQL statements in the given order: CREATE TABLE orders (order_id NUMBER(3) PRIMARY KEY, order_date DATE, customer_id number(3)); INSERT INTO orders VALUES (100,'10-mar-2007',222); ALTER TABLE orders MODIFY order_date NOT NULL; UPDATE orders SET customer_id=333; DELETE FROM order;

The DELETE statement results in the following error: ERROR at line 1: ORA-00942: table or view does not exist

What would be the outcome?

- A. All the statements before the DELETE statement would be rolled back.
B. All the statements before the DELETE statement would be implicitly committed within the session.
C. All the statements up to the ALTER TABLE statement would be committed and the outcome of UPDATE statement would be rolled back.
D. All the statements up to the ALTER TABLE statement would be committed and the outcome of the UPDATE statement is retained uncommitted within the session.

Answer: D

Question: 15

View the Exhibit and examine the structure of ORDER_ITEMS and ORDERS tables.

You need to remove from the ORDER_ITEMS table those rows that have an order status of 0 or 1 in the ORDERS table.

Which DELETE statements are valid? (Choose all that apply.)

- A. DELETE
FROM order_items
WHERE order_id IN (SELECT order_id
FROM orders
WHERE order_status in (0,1));
B. DELETE *

```
FROM order_items
WHERE order_id IN (SELECT order_id
FROM orders
WHERE order_status IN (0,1));
C. DELETE FROM order_items i
WHERE order_id = (SELECT order_id FROM orders o
WHERE i.order_id = o.order_id AND
order_status IN (0,1));
D. DELETE
FROM (SELECT * FROM order_items i,orders o
WHERE i.order_id = o.order_id AND order_status IN (0,1));
```

Answer: A, C, D