SET NULL

SQL

SQL schema

Structured Query Language (SQL)

SUM

UNION

UNIQUE constraint

UPDATE

VARIANCE

view

WHERE

Review Questions

7.1. The following table with purchase orders is created:

```
CREATE TABLE PURCHASE_ORDER
            (PONR CHAR(7) NOT NULL PRIMARY KEY,
            PODATE DATE,
            SUPNR CHAR(4) NOT NULL,
            FOREIGN KEY (SUPNR) REFERENCES
SUPPLIER (SUPNR)
               ON DELETE CASCADE ON UPDATE
CASCADE);
```

What happens upon deletion of a supplier?

a. All purchase order records tied to that supplier are also deleted.

b. The SUPNR of this supplier is replaced by a NULL value in PURCHASE_ORDER.

c. The SUPNR of this supplier is deleted in PURCHASE_ORDER.

d. The SUPNR of this supplier is only deleted in SUPPLIER.

7.2. We're interested in wine stores. Therefore, we want to retrieve the SUPNR and SUPNAME of each store which contains "wine" in its store name. Which of the following queries can we use?

a.

```
SELECT SUPNR, SUPNAME
FROM SUPPLIER
WHERE SUPNAME = "WINE"
```

b.

```
SELECT SUPNR, SUPNAME
FROM SUPPLIER
WHERE SUPNAME IS "%WINE%"
```

c.

```
SELECT SUPNR, SUPNAME
FROM SUPPLIER WHERE
SUPNAME LIKE "%WINE%"
```

d.

```
SELECT SUPNR, SUPNAME
FROM SUPPLIER
```

WHERE SUPNAME **IS** "WINE"

7.3. Take the following extract from SUPPLIES:

SUPNR	*PRODNR*	PURCHASE_PRICE	DELIV_PERIOD
37	0185	32.99	3
84	0185	33.00	5
94	0185	32.99	1

We want to retrieve the fastest delivery time for product 0185. We type the following query:

SELECT PRODNR, **MIN**(DELIV_PERIOD) **AS** MIN_DELIV_PERIOD
FROM SUPPLIES
WHERE PRODNR = '0185'

What are the results? If you believe the query is correct, select answer a, otherwise choose which results you believe will be retrieved.

a.

SUPNR	MIN_DELIV_PERIOD
94	1

b.

SUPNR	MIN_DELIV_PERIOD
37	3

c.

SUPNR	MIN_DELIV_PERIOD
37	1

d.

SUPNR	MIN_DELIV_PERIOD
37	1
84	1
94	1

7.4. Consider the following query:

SELECT *
FROM PRODUCT
WHERE PRODTYPE='red'
ORDER BY AVAILABLE_QUANTITY **DESC**,
PRODNAME

Which of the following answers is **correct**?

a.

PRODNR	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
474	Chateau De La Tour, Clos-Vougeot, Grand cru, 2008	red	147
885	Chateau Margaux, Grand Cru Classé, 1956	red	147
347	Chateau Corbin-Despagne, Saint-Emilion, 2005	red	145
832	Conde de Hervías, Rioja, 2004	red	121
…	…	…	…
331	Chateau La Commanderie, Lalande-de-Pomerol, 1998	red	3
219	Marques de Caceres, Rioja Crianza, 2010	red	0

b.

PRODNR	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
885	Chateau Margaux, Grand Cru Classé, 1956	red	147
474	Chateau De La Tour, Clos-Vougeot, Grand cru, 2008	red	147
347	Chateau Corbin-Despagne, Saint-Emilion, 2005	red	145
832	Conde de Hervías,Rioja, 2004	red	121
347	Chateau Corbin-Despagne, Saint-Emilion, 2005	red	145
832	Conde de Hervías, Rioja, 2004	red	121
…	…	…	…
331	Chateau La Commanderie, Lalande-de-Pomerol, 1998	red	3
219	Marques de Caceres, Rioja Crianza, 2010	red	0

c.

PRODNR	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
219	Marques de Caceres, Rioja Crianza, 2010	red	0
331	Chateau La Commanderie, Lalande-de-Pomerol, 1998	red	3
185	Chateau Petrus, 1975	red	5
523	Chateau Andron Blanquet, Saint Estephe, 1979	red	13
…	…	…	…
474	Chateau De La Tour, Clos-Vougeot, Grand cru, 2008	red	147
885	Chateau Margaux, Grand Cru Classé, 1956	red	147

d.

PRODNR	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
795	Casa Silva, Los Lingues, Carmenere, 2012	red	105
523	Chateau Andron Blanquet, Saint Estephe, 1979	red	13
977	Chateau Batailley, Grand Cru Classé, 1975	red	21
…	…	…	…
847	Seresin, Merlot, 1999	red	41
345	Vascosassetti, Brunello di Montalcino, 2004	red	64

7.5. We want to retrieve all unique supplier numbers and statuses of suppliers who have at least one outstanding purchase order. Which query is **correct**?

a.

```
SELECT DISTINCT R.SUPNR, R.SUPSTATUS
FROM SUPPLIER R, PURCHASE_ORDER O
```

b.

```
SELECT DISTINCT R.SUPNR, R.SUPSTATUS
FROM SUPPLIER R, PURCHASE_ORDER O
WHERE (R.SUPNR = O.SUPNR)
```

c.

```
SELECT DISTINCT R.SUPNR, R.SUPSTATUS
FROM SUPPLIER R, PURCHASE_ORDER O
WHERE (R.SUPNR = O.PONR)
```

d.

```
SELECT R.SUPNR, R.SUPSTATUS
FROM PURCHASE_ORDER R
```

7.6. Consider the following query:

```
SELECT P.PRODNR, P.PRODNAME,
P.AVAILABLE_QUANTITY, SUM(L.QUANTITY)
AS ORDERED_QUANTITY
FROM PRODUCT AS P LEFT OUTER JOIN PO_LINE
AS L
ON (P.PRODNR=L.PRODNR)
GROUP BY P.PRODNR
```

Which of the following statements is **not correct**?

a. The query retrieves the product number, product name, and available quantity of each product thanks to the left outer join.

b. The query retrieves for each product the total ordered quantity.

c. The query result can never contain NULL values.

d. If we remove the GROUP BY statement and P.PRODNR, P.PRODNAME, P.AVAILABLE_QUANTITY from the SELECT statement, the query will result in one row containing the total outstanding ordered quantity over all products in column "ORDERED_QUANTITY".

7.7. Consider following query:

```
SELECT DISTINCT P1.PRODNR, P1.PRODNAME
FROM PRODUCT P1, SUPPLIES S1
WHERE P1.PRODNR = S1.PRODNR AND
1 <= (SELECT COUNT(*) FROM SUPPLIES S2
    WHERE S2.SUPNR <> S1.SUPNR AND
P1.PRODNR=S2.PRODNR)
ORDER BY PRODNR
```

The query retrieves:

a. The number and name of all products that can only be supplied by one supplier.

b. The number and name of all products that cannot be supplied by any supplier.

c. The number and name of all products that can be supplied by more than one supplier.

d. The number and name of all products that can be supplied by all suppliers.

7.8. Which of the following queries selects the name of the supplier, corresponding order number, and total ordered quantity of the order that

has the maximum total quantity ordered.

a.

```
SELECT R1.SUPNAME, POL1.PONR,
SUM(POL1.QUANTITY)
FROM SUPPLIER R1, PURCHASE_ORDER PO1,
PO_LINE POL1
WHERE R1.SUPNR = PO1.SUPNR AND PO1.PONR =
POL1.PONR
GROUP BY POL1.PONR
HAVING SUM(POL1.QUANTITY) >= ANY
    (SELECT SUM(POL2.QUANTITY)
    FROM SUPPLIER R2, PURCHASE_ORDER PO2,
PO_LINE POL2
    WHERE R2.SUPNR = PO2.SUPNR AND PO2.PONR =
POL2.PONR
    GROUP BY POL2.PONR)
```

b.

```
SELECT R1.SUPNAME, POL1.PONR,
SUM(POL1.QUANTITY)
FROM SUPPLIER R1, PURCHASE_ORDER PO1,
PO_LINE POL1
WHERE R1.SUPNR = PO1.SUPNR AND PO1.PONR =
POL1.PONR
GROUP BY POL1.PONR
HAVING SUM(POL1.QUANTITY) <= ALL
        (SELECT SUM(POL2.QUANTITY)
        FROM SUPPLIER R2, PURCHASE_ORDER PO2,
PO_LINE POL2
        WHERE R2.SUPNR = PO2.SUPNR AND
PO2.PONR = POL2.PONR
```

```
        GROUP BY POL2.PONR)
```

c.

```
SELECT R1.SUPNAME, POL1.PONR,
SUM(POL1.QUANTITY)
FROM SUPPLIER R1, PURCHASE_ORDER PO1,
PO_LINE POL1
WHERE R1.SUPNR = PO1.SUPNR AND PO1.PONR =
POL1.PONR
GROUP BY POL1.PONR
HAVING SUM(POL1.QUANTITY) >= ALL
        (SELECT SUM(POL2.QUANTITY)
        FROM SUPPLIER R2, PURCHASE_ORDER PO2,
PO_LINE POL2
        WHERE R2.SUPNR = PO2.SUPNR AND
PO2.PONR = POL2.PONR
        GROUP BY POL2.PONR)
```

d.

```
SELECT R1.SUPNAME, POL1.PONR,
SUM(POL1.QUANTITY)
FROM SUPPLIER R1, PURCHASE_ORDER PO1,
PO_LINE POL1
WHERE R1.SUPNR = PO1.SUPNR AND PO1.PONR =
POL1.PONR
GROUP BY POL1.PONR
HAVING SUM(POL1.QUANTITY) <= ANY
        (SELECT SUM(POL2.QUANTITY)
        FROM SUPPLIER R2, PURCHASE_ORDER PO2,
PO_LINE POL2
        WHERE R2.SUPNR = PO2.SUPNR AND
PO2.PONR = POL2.PONR
```

```
        GROUP BY POL2.PONR)
```

7.9. Consider the following SQL query:

```
SELECT SUPNAME, SUPADDRESS, SUPCITY
FROM SUPPLIER R
WHERE NOT EXISTS
      (SELECT *
      FROM PRODUCT P
      WHERE EXISTS
            (SELECT *
            FROM SUPPLIES S
            WHERE R.SUPNR = S.SUPNR
            AND P.PRODNR = S.PRODNR));
```

This query selects:

a. The supplier name, supplier address, and supplier city of all suppliers who cannot supply any products.

b. The supplier name, supplier address, and supplier city of all suppliers who cannot supply all products.

c. The supplier name, supplier address, and supplier city of all suppliers who can supply at least one product.

d. The supplier name, supplier address, and supplier city of all suppliers who can supply all products.

7.10. Consider the following query:

```
SELECT P.PRODNR, P.PRODNAME
FROM PRODUCT P
```

```
WHERE EXISTS
        (SELECT *
        FROM PO_LINE POL
        WHERE P.PRODNR = POL.PRODNR
        GROUP BY POL.PRODNR
        HAVING SUM(POL.QUANTITY) >
P.AVAILABLE_QUANTITY)
```

The query retrieves:

a. The name and number of the product with the highest ordered quantity.

b. The name and number of all products that are ordered and do not exceed their available quantity.

c. The name and number of all products that are ordered and exceed their available quantity.

d. The name and number of the product with the lowest ordered quantity.

7.11. Consider following query:

```
SELECT CS.CURRENT_STOCK - O.ORDERED AS
NEW_STOCK
FROM (SELECT SUM(P.AVAILABLE_QUANTITY) AS
CURRENT_STOCK
        FROM PRODUCT P) AS CS,
        (SELECT SUM(POL.QUANTITY) AS ORDERED
        FROM PO_LINE POL) AS O
```

The output of the query represents:

a. A table summarizing for each product the increase in stock after the ordered products are delivered.

b. A table summarizing for each product the decrease in stock after the ordered products are delivered.

c. A scalar, summarizing the total quantity of products in stock after all the ordered products are delivered.

d. A scalar, summarizing the decrease in total available quantity of all products after the ordered products are delivered.

7.12. Given the task to retrieve the numbers of all suppliers who can supply products 0832 and 0494, which query is **correct**?

a.

```
SELECT DISTINCT SUPNR
FROM SUPPLIES
WHERE PRODNR IN (0832, 0494)
```

b.

```
SELECT SUPNR
FROM SUPPLIES
WHERE PRODNR = 0832
UNION ALL
SELECT SUPNR
FROM SUPPLIES
WHERE PRODNR = 0494
```

c.

```
SELECT SUPNR
```

FROM SUPPLIES
WHERE PRODNR = 0832
INTERSECT
SELECT SUPNR
FROM SUPPLIES
WHERE PRODNR = 0494

d.

SELECT UNIQUE SUPNR
FROM SUPPLIES
WHERE PRODNR **IN** (0832, 0494)

7.13. Consider the following View definition and update statement:

CREATE VIEW
TOPPRODUCTS(PRODNR,PRODNAME,QUANTITY) **AS**
SELECT PRODNR, PRODNAME,
AVAILABLE_QUANTITY
FROM PRODUCT **WHERE**
AVAILABLE_QUANTITY>100
WITH CHECK OPTION
UPDATE TOPPRODUCTS
SET QUANTITY=80
WHERE PRODNR=0153

What will be the result of this?

a. The update can be successfully made but only the PRODUCT table will be updated.

b. The update can be successfully made and both the View and PRODUCT table will be updated.

c. The update will be halted because of the WITH CHECK OPTION.

d. The update can be successfully made but only the View will be updated.

7.14. Compare the following two queries:

1.

```
SELECT COUNT(DISTINCT SUPNR)
FROM PURCHASE_ORDER
```

2.

```
SELECT COUNT(SUPNR)
FROM PURCHASE_ORDER
```

Which of the following statements is correct?

a. Result query 1 is always = result query 2 because PURCHASE_ORDER contains only unique purchase orders.

b. Result query 1 is always ≤ result query 2 because the DISTINCT operator counts only unique SUPNRs.

c. Result query 1 is always ≥ result query 2 because query 1 sums the number of purchase orders per supplier while query 2 sums the number of purchase orders in total.

d. Result query 1 is sometimes ≥ and sometimes ≤ result query 2 because the result depends on the number of suppliers and the number of purchase orders.

7.15. Consider the following query:

```
SELECT PRODNR, AVG(QUANTITY) AS
AVG_QUANTITY
FROM PO_LINE
GROUP BY PRODNR
HAVING SUM(QUANTITY) < 15
```

What is the result?

a. The query returns the PRODNR and average QUANTITY of each purchase order that has fewer than 15 purchase order lines.

b. The query returns the PRODNR and average QUANTITY of each product that has fewer than 15 purchase order lines.

c. The query returns the PRODNR and average QUANTITY of each product that has fewer than 15 orders.

d. The query returns the PRODNR and average QUANTITY of each purchase order that has fewer than 15 orders.

7.16. Consider the following query:

```
SELECT PRODNAME
FROM PRODUCT
WHERE PRODNR IN
    (SELECT PRODNR
    FROM SUPPLIES
    WHERE SUPNR IN
        (SELECT SUPNR
        FROM SUPPLIER
        WHERE SUPCITY = 'New York'))
```

```
  AND PRODNR IN
(SELECT PRODNR
FROM SUPPLIES
WHERE SUPNR IN
      (SELECT SUPNR
      FROM SUPPLIER
      WHERE SUPCITY = 'Washington'))
```

What is the result?

a. The query retrieves the product name of each product that has a supplier in New York or Washington.

b. The query retrieves the product name of each product that has both a supplier in New York and a supplier in Washington.

c. The query retrieves the product name of each product along with all possible supplier cities.

d. The query incorrectly combines every product name and supplier city.

7.17. We want to retrieve the available quantity of each ordered product of supplier Ad Fundum. Which of the following queries is correct?

a.

```
SELECT PRODNR, AVAILABLE_QUANTITY
FROM PRODUCT
WHERE PRODNR IN
      (SELECT PRODNR
      FROM PO_LINE) AND
SUPNR IN
      (SELECT SUPNR
```

```
        FROM SUPPLIER
        WHERE SUPNAME='Ad Fundum')
```

b.

```
SELECT PRODNR, AVAILABLE_QUANTITY
FROM PRODUCT
WHERE SUPNR IN
        (SELECT SUPNR
        FROM SUPPLIER
        WHERE SUPNAME='Ad Fundum')
```

c.

```
SELECT PRODNR, AVAILABLE_QUANTITY
FROM PRODUCT
WHERE PRODNR IN
        (SELECT PRODNR
        FROM PO_LINE
        WHERE PONR IN
          (SELECT PONR
          FROM PURCHASE_ORDER
          WHERE SUPNR IN
              (SELECT SUPNR
              FROM SUPPLIER
              WHERE SUPNAME='Ad Fundum')))
```

d.

```
SELECT PRODNR, AVAILABLE_QUANTITY
FROM PRODUCT
WHERE PRODNR =
         (SELECT PRODNR
         FROM PO_LINE
```

```
        WHERE PONR =
          (SELECT PONR
          FROM PURCHASE_ORDER
          WHERE SUPNR =
            (SELECT SUPNR
            FROM SUPPLIER
            WHERE SUPNAME='Ad Fundum')))
```

7.18. Consider the following SQL query:

```
SELECT P1.PRODNR
    FROM PRODUCT P1
    WHERE 5 <=
        (SELECT COUNT(*)
        FROM PRODUCT P2
        WHERE P1.PRODNR < P2.PRODNR)
```

This query selects:

a. The five highest product numbers.

b. The five lowest product numbers.

c. All product numbers except for the five lowest product numbers.

d. All product numbers except for the five highest product numbers.

7.19. Consider the following query:

```
SELECT R1.SUPNAME, R1.SUPNR, COUNT(*)
FROM PURCHASE_ORDER PO1, SUPPLIER R1
WHERE PO1.SUPNR = R1.SUPNR
GROUP BY R1.SUPNR
HAVING COUNT(*) >= ALL
```

```
(SELECT COUNT(*)
FROM PURCHASE_ORDER PO2, SUPPLIER R2
WHERE PO2.SUPNR = R2.SUPNR
GROUP BY R2.SUPNR)
```

The query retrieves:

a. The name, number, and total outstanding orders of all suppliers that have outstanding orders.

b. The name, number, and total outstanding orders of all suppliers that have outstanding orders, except for the supplier(s) with the fewest outstanding orders.

c. The name, number, and total outstanding orders of the supplier with the most outstanding orders.

d. The name, number, and total outstanding orders of the supplier with the fewest outstanding orders.

7.20. Consider the following query:

```
SELECT P.PRODNR, P.PRODNAME
FROM PRODUCT P
EXCEPT
SELECT POL.PRODNR
FROM PO_LINE POL
```

The query retrieves:

a. The number and name of all the products with no outstanding order.

b. The number and name of all the products that are ordered.

c. The query will not execute because both queries do not select the same columns.

d. The query will not execute because both queries do not select the same rows.

7.21. Consider following query:

```
SELECT P1.PRODNR, P1.PRODNAME, S1.SUPNR,
S1.PURCHASE_PRICE
FROM PRODUCT P1, SUPPLIES S1
WHERE P1.PRODNR = S1.PRODNR
AND NOT EXISTS
    (SELECT *
    FROM PRODUCT P2, SUPPLIES S2
    WHERE P2.PRODNR = S2.PRODNR
    AND P1.PRODNR = P2.PRODNR
    AND S1.PURCHASE_PRICE > S2.PURCHASE_PRICE)
```

and the following statements:

1. For each product, the supplier number of the supplier who can supply the product for the cheapest price is retrieved.

2. For each product, the supplier number of the supplier who supplies the product for the highest price is retrieved.

3. For each product, exactly one tuple is returned.

4. For each product, more than one tuple can be returned.

Which statements are true?

a. 1 and 3.

b. 1 and 4.

c. 2 and 3.

d. 2 and 4.

7.22. Consider the following query:

```
SELECT R.SUPNAME, (SELECT COUNT(PO.PODATE)
                   FROM PURCHASE_ORDER PO
                   WHERE R.SUPNR = PO.SUPNR)
AS SUMMARY
FROM SUPPLIER R
```

The query selects:

a. The name and total number of outstanding orders of all suppliers that have at least one outstanding order.

b. The name and total number of outstanding orders of all suppliers.

c. The supplier name and order date of each of his/her outstanding orders.

d. The supplier name and order date of each of his/her outstanding orders. If a supplier does not have an outstanding order, she/he will be included in the output with a null value for the “SUMMARY” column.