Trial Exam May 2024

Started on	Tuesday, 7 May 2024, 08:00
State	Finished
Completed on	Tuesday, 7 May 2024, 11:54
Time taken	3 hours 54 mins

Information
Flag question

Instructions:

- You have 4 hours to answer 7 problems described in the following sections. Note: if you have been granted special exam conditions by SAP, you might have extra time.
- Except for section 3, all sections have a preamble with the general instructions for that particular section.
- You must provide all your answers online, i.e., directly in this Exam Quiz.

Negative points: in all multiple-choice questions, a wrong answer gives negative points so that the sum of all wrong answers cancels out the correct answers. For example, if you have 5 options, two are correct and three are wrong, each correct answer is worth 50% and each wrong answer is worth -33.33%. The lowest mark you can get from a question is 0, so if your wrong choices percentage exceeds the correct choices it will result in 0.

IMPORTANT: it is your responsibility to ensure that you FINISH and SUBMIT your attempt: if an exam is not properly submitted we WILL NOT take it into account. You have **only one exam attempt**. After you SUBMIT the quiz, you cannot start a new attempt anymore.

Good luck!

Information
Flag question

In this exam, you will work with the Northwind (northwind.sql) database using your PostgreSQL installation (remember to load the data using *psql*). The database contains sales data for a fictitious company called Northwind Traders, which imports and exports specialty foods from around the world.

The following 11 tables exist in the Northwind database and are relevant to this exam:

1. Categories(category_id, category_name, description, picture): list of product categories. 8 tuples.
2. Suppliers(supplier_id, company_name, contact_name, contact_title, address, city, region, postal_code, country, phone, fax, homepage): suppliers of Northwind Traders. 29 tuples.
3. Products(product_id, product_name, supplier_id, category_id, quantity_per_unit, unit_price, units_in_stock, units_on_order, reorder_level, discontinued): the products Northwind Traders buys and sells. 77 tuples.
4. Customers(customer_id, company_name, contact_name, contact_title, address, city, region, postal_code, country, phone, fax): buyers of Northwind Traders products. 91 tuples.
5. Employees(employee_id, last_name, first_name, title, title_of_courtesy, birth_date, hire_date, address, city, region, postal_code, country, home_phone, extension, photo, notes, reports_to, photo_path): people who work for Northwind Traders. 9 tuples.
6. Shippers(shipper_id, company_name, phone): companies who ship products from Northwind Traders to Northwind Traders' customers. 6 tuples.
7. Orders(order_id, customer_id, employee_id, order_date, required_date, shipped_date, ship_via, freight, ship_name, ship_address, ship_city, ship_region, ship_postal_code, ship_country): information about each order such as the customer who placed the order, the employee who handled the order, the order and shipment date, which shipper made the shipment, etc. 830 tuples.
8. Order_details(order_id, product_id, unit_price, quantity, discount): transaction details of each order such as products, unit price and quantity, and discount applied. 2155 tuples.
9. Region(region_id, region_description): regions from which the Northwind company's employees operate. 4 tuples.
10. Territories(territory_id, territory_description, region_id): territories within regions. 53 tuples.
11. Employee_territories(employee_id, territory_id): assignment of working territories to employees. 49 tuples.

To understand the details of the tables, you may also study the DDL commands (the CREATE TABLE statements are at the top of the script) or inspect the tables using SQL queries.

Information
Flag question

[Question for BSc only]

To answer this question, you will need to study the schema of the Northwind database in the "Database Description" section.

Consider the following code:

```
DROP TRIGGER IF EXISTS OfferDiscountUS ON Order_details;
DROP FUNCTION IF EXISTS OfferDiscountUS;

CREATE OR REPLACE FUNCTION OfferDiscountUS()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $BODY$
DECLARE country VARCHAR(250);
BEGIN
  -- Get country of customer
  SELECT c.country INTO country
  FROM Orders o
  JOIN Customers c ON o.customer_id = c.customer_id
  WHERE o.order_id = NEW.order_id;

  -- Update discount if the customer is from the US
  IF country = 'USA' THEN
    NEW.discount = NEW.discount + 0.3;
  END IF;

  RETURN NEW;
END;
$BODY$;

CREATE TRIGGER OfferDiscountUS
AFTER INSERT ON Order_details
FOR EACH ROW EXECUTE FUNCTION OfferDiscountUS();
```

Question 1
Not answered
Marked out of 5.00
Flag question

[Question for BSc only]

Given the preamble above, select the true statements:

- ☐ a. The trigger will affect both inserts and updates
- ☐ b. NEW is not available in row-level AFTER triggers
- ☐ c. If there are multiple inserts to Order_details in a transaction, the trigger will be fired each time
- ☐ d. The trigger is correctly written as an AFTER trigger
- ☐ e. This trigger will be activated after an insert to Order_details

Information
Flag question

[Question for MSc only]

To answer this question, you will need to study the schema of the Northwind database in the "Database Description" section.

Consider the following Python code:

```
def checkAndUpdateUnitPrice(conn: pg.Connection, product_id: int, new_unit_price: float):
  cur = None
  try:
    conn.autocommit = False
    cur = conn.cursor()
    cur.execute("""
          SELECT unit_price
          FROM Products
          WHERE product_id = %s
          """, [product_id])
    current_unit_price = cur.fetchone()[0]

    if current_unit_price < new_unit_price:
      cur.execute("""
            UPDATE Products
            SET unit_price = %s
            WHERE product_id = %s
            """, [product_id, new_unit_price])

    cur.execute("""
          SELECT unit_price
          FROM Products
          WHERE product_id = %s
          """, [product_id])
    check_unit_price = cur.fetchone()[0]
    conn.commit()
    return check_unit_price
  except Exception as error:
    conn.rollback()
    raise error
  finally:
    if cur is not None:
      cur.close()
```

Question 2
Complete
Marked out of 5.00
Flag question

[Question for MSc only]

Given the preamble above, select the true statements:

- ☐ a. The connection automatically resets autocommit to True after exiting the function
- ☑ b. This function is protected against SQL Injections
- ☑ c. If the current unit price of the given product is less than the new unit price, the unit price of the product is set to the new unit price
- ☑ d. If an exception is raised, the cursor is safely closed
- ☐ e. If this function runs successfully, the finally block is not run due to the return in the try block

Information
Flag question

In the first part of this section, you have to select the SQL query corresponding to each given question formulated in English.

In the second part of this section, you have to write an SQL query for each given question formulated in English.

In both parts, you should also enter the query's numerical answer.

Requirements: Queries must return correct results for any database instance. Queries should not return anything except the answer; a query that returns more information will not receive full points, even if the answer is part of the returned result. A sequence of several queries that answer the question will not receive full points, but subqueries and views can be used if needed. Queries should be as simple as possible; queries that are unnecessarily complex may not get full points, despite returning the correct answer.

Question 3
Complete
Marked out of 4.00
Flag question

How many suppliers are not from the USA?

○ a.

```
SELECT *
FROM Suppliers
WHERE country <> 'USA';
```

○ b.

```
SELECT COUNT(*)
FROM Suppliers
WHERE country != 'USA';
```

○ c.

```
SELECT COUNT(*)
FROM Suppliers
WHERE region IS NULL;
```

○ d.

```
SELECT COUNT(*)
FROM Suppliers
WHERE country NOT LIKE '%USA%';
```

● e.

```
SELECT COUNT(*)
FROM Suppliers
WHERE country <> 'USA';
```

○ f.

```
SELECT COUNT(*)
FROM Suppliers
WHERE region <> 'US';
```

Question 4
Complete
Marked out of 1.00
Flag question

How many suppliers are not from the USA?

Answer: 25

Question 5
Complete
Marked out of 4.00
Flag question

How many products have been sold by suppliers from Tokyo?

○ a.

```
SELECT COUNT(*)
FROM Order_details od
JOIN Products p ON od.product_id = p.product_id
WHERE p.supplier_id = 4;
```

○ b.

```
SELECT COUNT(*)
FROM Products p
JOIN Suppliers s ON p.supplier_id = s.supplier_id
WHERE s.city = 'Tokyo';
```

● c.

```
SELECT COUNT(*)
FROM Order_details od
JOIN Products p ON od.product_id = p.product_id
JOIN Suppliers s ON p.supplier_id = s.supplier_id
WHERE s.city = 'Tokyo';
```

○ d.

```
SELECT COUNT(*)
FROM Order_details od
JOIN Products p ON od.product_id = p.product_id
WHERE p.supplier_id = (
    SELECT supplier_id
    FROM Suppliers
    WHERE city = 'Tokyo'
);
```

○ e.

```
SELECT COUNT(*)
FROM Order_details od
JOIN Products p ON od.product_id = p.product_id
JOIN Suppliers s ON p.supplier_id = p.supplier_id
WHERE s.city = 'Tokyo';
```

○ f.

```
SELECT COUNT(*)
FROM Order_details od
JOIN Products p ON od.product_id = p.product_id
WHERE p.supplier_id IN (
    SELECT supplier_id
    FROM Suppliers
    WHERE city = 'Tokyo'
);
```

Question 6
Complete
Marked out of 1.00
Flag question

How many products have been sold by suppliers from Tokyo?

Answer: 51

Question 7
Complete
Marked out of 4.00
Flag question

What is the maximum number of suppliers that supply products in the same order?

● a.

```
SELECT MAX(supplier_count)
FROM (
    SELECT od.order_id, COUNT(DISTINCT p.supplier_id) AS supplier_count
    FROM Order_details od
    JOIN Products p ON od.product_id = p.product_id
    GROUP BY od.order_id
) AS order_supplier_counts;
```

○ b.

```
SELECT MAX(supplier_count)
FROM (
    SELECT o.order_id, COUNT(DISTINCT p.supplier_id) AS supplier_count
    FROM Orders o
    JOIN Order_details od ON o.order_id = od.order_id
    JOIN Products p ON od.product_id = p.product_id
    GROUP BY o.order_id
) AS order_supplier_counts;
```

○ c.

```
SELECT MAX(supplier_count)
FROM (
    SELECT COUNT(p.supplier_id) AS supplier_count
    FROM Order_details od
    JOIN Products p ON od.product_id = p.product_id
    GROUP BY p.supplier_id
) AS order_supplier_counts;
```

○ d.

```
SELECT MAX(supplier_count)
FROM (
    SELECT COUNT(p.supplier_id) AS supplier_count
    FROM Order_details od
    JOIN Products p ON od.product_id = p.product_id
    GROUP BY p.product_id
) AS order_supplier_counts;
```

○ e.

```
SELECT COUNT(supplier_max)
FROM (
    SELECT od.order_id, MAX(p.supplier_id) AS supplier_max
    FROM Order_details od
    JOIN Products p ON od.product_id = p.product_id
    GROUP BY od.order_id
) AS order_supplier_counts;
```

○ f.

```
SELECT MAX(supplier_count)
FROM (
    SELECT od.order_id, COUNT(p.supplier_id) AS supplier_count
    FROM Order_details od
    JOIN Products p ON od.product_id = p.product_id
    GROUP BY od.order_id
) AS order_supplier_counts;
```

Question 8
Complete
Marked out of 1.00
Flag question

What is the maximum number of suppliers that supply products in the same order?

Answer: 18

Question 9
Complete
Marked out of 4.00
Flag question

Which orders only contain suppliers from a single country?

○ a.

```
SELECT o.order_id
FROM Orders o
JOIN Order_details od ON o.order_id = od.order_id
JOIN Products p ON od.product_id = p.product_id
JOIN Suppliers s ON p.supplier_id = s.supplier_id
GROUP BY o.order_id
HAVING COUNT(DISTINCT s.country) = 1;
```

○ b.

```
SELECT o.order_id, o.customer_id
FROM Orders o
JOIN Order_details od ON o.order_id = od.order_id
JOIN Products p ON od.product_id = p.product_id
JOIN Suppliers s ON p.supplier_id = s.supplier_id
GROUP BY o.order_id, o.customer_id
HAVING COUNT(DISTINCT s.country) = 1;
```

○ c.

```
SELECT s.supplier_id
FROM Order_details od
JOIN Products p ON od.product_id = p.product_id
JOIN Suppliers s ON p.supplier_id = s.supplier_id
GROUP BY s.supplier_id
HAVING COUNT(DISTINCT s.country) = 1;
```

○ d.

```
SELECT o.order_id, o.customer_id
FROM Orders o
JOIN Order_details od ON o.order_id = od.order_id
JOIN Products p ON od.product_id = p.product_id
JOIN Suppliers s ON p.supplier_id = s.supplier_id
GROUP BY o.order_id, o.customer_id
HAVING COUNT(s.country) = 1;
```

○ e.

```
SELECT od.order_id
FROM Order_details od
JOIN Products p ON od.product_id = p.product_id
JOIN Suppliers s ON p.supplier_id = s.supplier_id
GROUP BY od.order_id
HAVING COUNT(s.supplier_id) = 1;
```

● f.

```
SELECT od.order_id
FROM Order_details od
JOIN Products p ON od.product_id = p.product_id
JOIN Suppliers s ON p.supplier_id = s.supplier_id
GROUP BY od.order_id
HAVING COUNT(DISTINCT s.country) = 1;
```

Question 10
Complete
Marked out of 1.00
Flag question

How many results does the previous query return?

Note: you should report the NUMBER OF RESULTS, not the result itself.

Answer: 167

Question 11
Complete
Marked out of 4.00
Flag question

How many employees operate from all regions of Northwind?

Write your SQL Query here:

```
-- 11 How many employees operate from all regions of Northwind?
-- Division query

SELECT count(DISTINCT e.employee_id)
FROM employees e
          JOIN employee_territories et ON e.employee_id = et.employee_id
          JOIN territories t ON et.territory_id = t.territory_id
          JOIN region r ON t.region_id = r.region_id
GROUP BY e.employee_id
HAVING count(DISTINCT r.region_id) = (SELECT count(DISTINCT r.region_id) FROM region r);

-- To test the query, we insert the necessary data into the DB. It correctly identifies 1 employee if run the following DML against the DB insta
INSERT INTO employees
VALUES (11, 'TESTUSER', 'Nancy', 'Sales Representative', 'Ms.', '1948-12-08', '1992-05-01',
          '507 - 20th Ave. E.\nApt. 2A', 'Seattle', 'WA', '98122', 'USA', '(206) 555-9857', '5467', '\x',
          'Education includes a BA in psychology from Colorado State University in 1970.  She also completed The Art of the Cold Call.  Na
          2, 'http://accweb/emmployees/davolio.bmp');
INSERT INTO employee_territories
VALUES (10, '01581');
INSERT INTO employee_territories
```

Question 12
Complete
Marked out of 1.00
Flag question

How many employees operate from all regions of Northwind?

Answer: 0

Question 13
Complete
Marked out of 4.00
Flag question

There are 22 orders with fewer than 30 products (quantity included) that have a freight above 50.0.

How many orders are there with 10 or fewer products (quantity included) that have a freight that is more than 10.0?

Write your SQL Query here:

```
-- 12 There are 22 orders with fewer than 30 products (quantity included) that have a freight above 50.0.
-- How many orders are there with 10 or fewer products (quantity included) that have a freight that is more than 10.0?

SELECT count(DISTINCT o.order_id)
FROM (SELECT order_id, SUM(quantity) AS total_quantity
        FROM order_details
        GROUP BY order_id
        HAVING SUM(quantity) <= 10) AS sub
            JOIN orders o ON sub.order_id = o.order_id
WHERE o.freight > 10.0;
```

Question 14
Complete
Marked out of 1.00
Flag question

How many orders are there with 10 or fewer products (quantity included) that have a freight that is more than 10.0?

Answer: 13

Question 15
Complete
Marked out of 4.00
Flag question

How many customers have a contact title that ends in "Manager"?

Write your SQL Query here:

```
-- 15 How many customers have a contact title that ends in "Manager"?

SELECT count(DISTINCT customer_id)
FROM customers
WHERE contact_title LIKE '%Manager';
```

Question 16
Complete
Marked out of 1.00
Flag question

What is the result of the previous query?

Answer: 33

Question 17
Complete
Marked out of 4.00
Flag question

How many employees have handled orders for products in all categories?

Write your SQL query here:

```
-- 17 How many employees have handled orders for products in all categories?

SELECT count(DISTINCT e.employee_id)
FROM employees e
          JOIN orders o ON o.employee_id = e.employee_id
WHERE o.order_id = (SELECT order_id
                              FROM order_details
                                        JOIN public.products p ON order_details.product_id = p.product_id
                                        JOIN public.categories c ON p.category_id = c.category_id
                              GROUP BY order_id
                              HAVING count(DISTINCT c.category_id) = (SELECT count(*) FROM categories))
```

Question 18
Complete
Marked out of 1.00
Flag question

What is the result of the previous query?

Answer: 1

Question 19
Complete
Marked out of 4.00
Flag question

Consider a table Employee(ID, Name, Department, Office) with the following dependencies:

- ID → Name
- ID,Name → Department
- Department → Office
- Department,Office → Office

Normalize Employee to BCNF and select the resulting relations:

- ☐ a. Employee1(ID,Department,Office)
- ☑ b. Employee2(Department,Office)
- ☑ c. Employee3(ID,Name,Department)
- ☐ d. Employee4(ID,Name)
- ☐ e. Employee5(ID,Name,Office)
- ☐ f. Employee6(ID,Office)

Question 20
Complete
Marked out of 3.00
Flag question

Consider a relation R(A,B,C,D,E) with the following dependencies:

- C → A
- A → C
- E → D
- C → BDE

Select the true statements:

- ☑ a. Normalizing to BCNF results in exactly two relations.
- ☐ b. C is the only key of R.
- ☐ c. In a distributed NoSQL system, normalizing R to BCNF will improve performance.
- ☐ d. The relation Z(A,B,C,D) is in BCNF.
- ☐ e. DE → D is a trivial functional dependency.

Question 21
Complete
Marked out of 3.00
Flag question

Consider a relation Book(ISBN,Title,Author,Genre,PublicationYear) with the following dependencies:

- ISBN → Title,Author,Genre,PublicationYear
- Title,Author → PublicationYear
- Title,Author → ISBN

Select the true statements:

- ☐ a. The relation Book is in BCNF.
- ☑ b. ISBN is the only key of the Book relation.
- ☐ c. Title,Author,Genre → PublicationYear is an unavoidable functional dependency.
- ☐ d. Normalizing to BCNF results in exactly two relations.

Information
Flag question

Consider the following schedule of operations seen by a database management system (time flows from left to right):

Transaction 1:	**R(A)**		**R(D)**		**W(D)**	**R(C)**	**W(C)**
Transaction 2:		**R(B)**			**R(C)**	**W(B)**	
Transaction 3:			**R(C)**	**W(C)**			
Transaction 4:		**R(B)**		**R(D)**		**R(C)**	**W(B)**

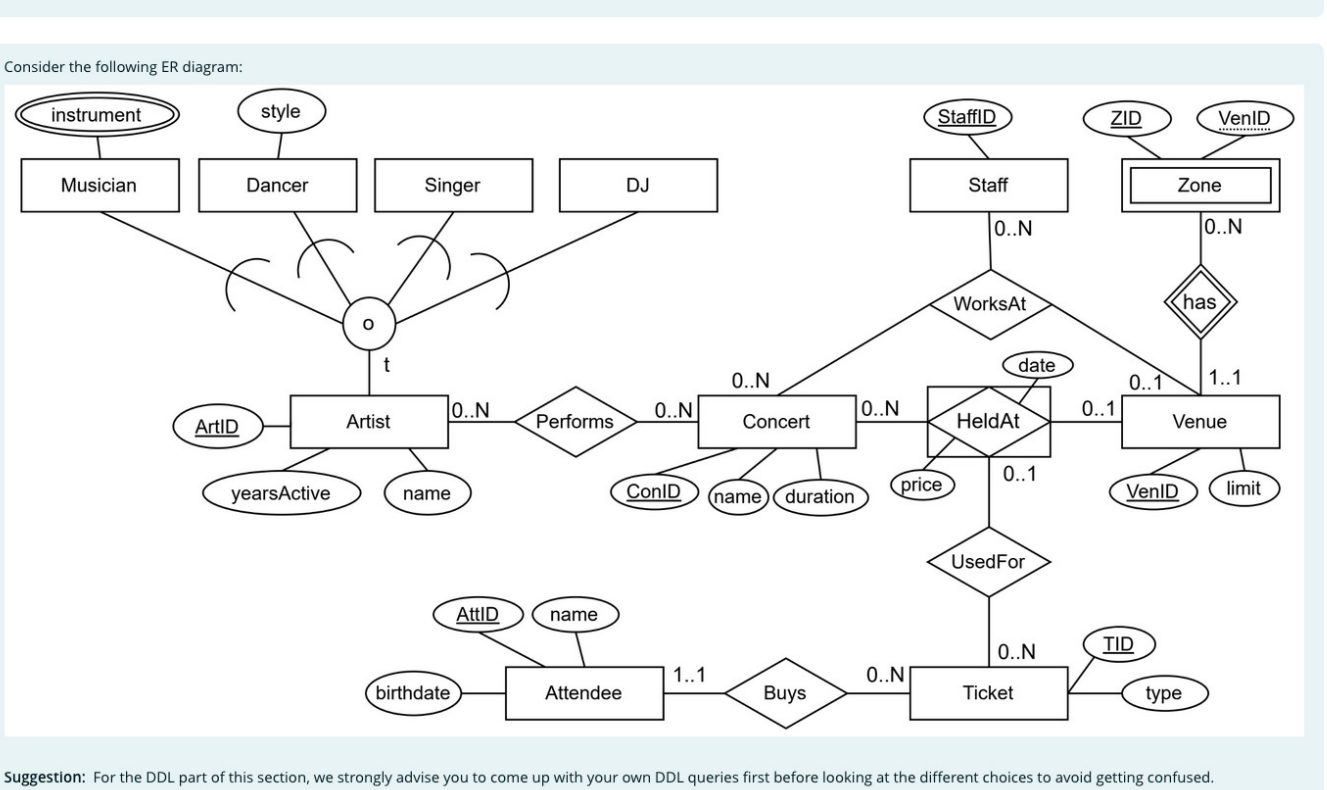
Question 22
Complete
Marked out of 5.00
Flag question

Given the preamble above, select the true statements when the database management system uses Rigorous-2PL:

- ☑ a. Transaction 3 finishes first
- ☑ b. Transaction 4 blocks Transaction 1
- ☐ c. Transaction 2 will run to completion
- ☐ d. Transaction 2 and 1 run into a deadlock
- ☐ e. If Transaction 1 is aborted the deadlock is resolved

Information
Flag question

Consider the following ER diagram:



Suggestion: For the DDL part of this section, we strongly advise you to come up with your own DDL queries first before looking at the different choices to avoid getting confused.

Question 23
Complete
Marked out of 10.00
Flag question

Given the ER Diagram in the preamble, select the true statements:

- ☐ a. An Artist can be a Singer and a Dancer but not a DJ
- ☐ b. An Artist can exist without any specialty
- ☑ c. An Attendee can buy many Tickets
- ☐ d. A Ticket can be bought by multiple Attendees
- ☐ e. A Staff member working at a Concert can work at multiple Venues
- ☐ f. A Ticket can be used for multiple Concerts
- ☑ g. A Zone belongs to a specific Venue
- ☑ h. A Staff member working at a Venue can work at many Concerts
- ☑ i. A Ticket can be used for maximum one Concert being held at a Venue
- ☑ j. A Concert can be held at maximum one Venue

Question 24
Complete
Marked out of 2.00
Flag question

Select the best DDL query to create the Artist relation in the ER Diagram. The relations must include all key and foreign key constraints. Make reasonable assumptions about the attributes.

● a.

```
CREATE TABLE Artist (
    ArtID SERIAL PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    yearsActive INTEGER NOT NULL
);
```

○ b.

```
CREATE TABLE Artist (
    ArtID SERIAL PRIMARY KEY,
    name VARCHAR(50),
    yearsActive INTEGER
);
```

○ c.

```
CREATE TABLE Artist (
    ArtID SERIAL,
    name VARCHAR(50) NOT NULL,
    yearsActive DATE
);
```

○ d.

```
CREATE TABLE Artist (
    ArtID SERIAL,
    name VARCHAR(50),
    yearsActive INTEGER
);
```

Question 25
Complete
Marked out of 2.00
Flag question

Select the best DDL query to create the Musician relation in the ER Diagram. The relations must include all key and foreign key constraints. Make reasonable assumptions about the attributes.

○ a.

```
CREATE TABLE Musician (
    ArtID INTEGER REFERENCES Artist(ArtID),
    PRIMARY KEY (ArtID)
);

CREATE TABLE Instruments (
    ArtID INTEGER REFERENCES Artist(ArtID),
    instrument VARCHAR (255) NOT NULL,
    PRIMARY KEY (ArtID, instrument)
);
```

○ b.

```
CREATE TABLE Musician (
    ArtID INTEGER REFERENCES Artist(ArtID),
    instruments VARCHAR(255) NOT NULL,
    PRIMARY KEY (ArtID)
);
```

○ c.

```
CREATE TABLE Musician (
    ArtID INTEGER REFERENCES Artist(ArtID),
    MID INTEGER NOT NULL,
    instruments VARCHAR(255) NOT NULL,
    PRIMARY KEY (ArtID, MID)
);
```

● d.

```
CREATE TABLE Musician (
    ArtID INTEGER REFERENCES Artist(ArtID),
    PRIMARY KEY (ArtID)
);

CREATE TABLE Instruments (
    ArtID INTEGER REFERENCES Musician(ArtID),
    instrument VARCHAR(255) NOT NULL,
    PRIMARY KEY (ArtID, instrument)
);
```

Question 26
Complete
Marked out of 2.00
Flag question

Select the best DDL query to create the Concert relation in the ER Diagram. The relations must include all key and foreign key constraints. Make reasonable assumptions about the attributes.

○ a.

```
CREATE TABLE Concert (
    ConID SERIAL PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    duration INTEGER NOT NULL,
    heldDate DATE,
    ticketPrice FLOAT,
    VenID INTEGER REFERENCES Venue(VenID),
    UNIQUE (ConID, VenID)
);
```

○ b.

```
CREATE TABLE Concert (
    ConID SERIAL PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    duration INTEGER NOT NULL,
    heldDate DATE NOT NULL,
    ticketPrice FLOAT NOT NULL,
    VenID INTEGER NOT NULL REFERENCES Venue(VenID)
);
```

● c.

```
CREATE TABLE Concert (
    ConID SERIAL PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    duration INTEGER NOT NULL
);
```

○ d.

```
CREATE TABLE Concert (
    ConID SERIAL PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    duration INTEGER NOT NULL,
    VenID INTEGER REFERENCES Venue(VenID)
);
```

Question 27
Complete
Marked out of 2.00
Flag question

Select the best DDL query to create the Ticket relation in the ER Diagram. The relations must include all key and foreign key constraints. Make reasonable assumptions about the attributes.

○ a.

```
CREATE TABLE Ticket (
    TID SERIAL PRIMARY KEY,
    type INTEGER NOT NULL,
    AttID INTEGER NOT NULL REFERENCES Attendee(AttID),
    ConID INTEGER,
    VenID INTEGER,
    FOREIGN KEY (ConID, VenID) REFERENCES Concert(ConID, VenID)
);
```

● b.

```
CREATE TABLE Ticket (
    TID SERIAL PRIMARY KEY,
    type INTEGER NOT NULL,
    AttID INTEGER NOT NULL REFERENCES Attendee(AttID),
    ConID INTEGER,
    VenID INTEGER,
    FOREIGN KEY (ConID, VenID) REFERENCES HeldAt(ConID, VenID)
);
```

○ c.

```
CREATE TABLE Ticket (
    TID SERIAL PRIMARY KEY,
    type INTEGER NOT NULL,
    AttID INTEGER NOT NULL REFERENCES Attendee(AttID),
    ConID INTEGER REFERENCES Concert(ConID),
    VenID INTEGER REFERENCES Venue(VenID)
);
```

○ d.

```
CREATE TABLE Ticket (
    TID SERIAL PRIMARY KEY,
    type INTEGER NOT NULL,
    AttID INTEGER REFERENCES Attendee(AttID),
    ConID INTEGER,
    VenID INTEGER,
    FOREIGN KEY (ConID, VenID) REFERENCES HeldAt(ConID, VenID)
);
```

Question 28
Complete
Marked out of 2.00
Flag question

Select the best DDL query to create the WorksAt relationship in the ER Diagram. The relations must include all key and foreign key constraints. Make reasonable assumptions about the attributes.

○ a.

```
CREATE TABLE WorksAt (
    StaffID INTEGER REFERENCES Staff(StaffID),
    ConID INTEGER REFERENCES Concert(ConID),
    VenID INTEGER NOT NULL REFERENCES Venue(VenID),
    PRIMARY KEY (StaffID, ConID)
);
```

○ b.

```
CREATE TABLE WorksAt (
    StaffID INTEGER REFERENCES Staff(StaffID),
    ConID INTEGER NOT NULL REFERENCES Concert(ConID),
    VenID INTEGER REFERENCES Venue(VenID),
    PRIMARY KEY (StaffID, VenID)
);
```

● c.

```
CREATE TABLE WorksAt (
    StaffID INTEGER REFERENCES Staff(StaffID),
    ConID INTEGER REFERENCES Concert(ConID),
    VenID INTEGER REFERENCES Venue(VenID),
    PRIMARY KEY (StaffID, ConID, VenID)
);
```

○ d.

```
CREATE TABLE Staff (
    StaffID SERIAL PRIMARY KEY,
    VenID INTEGER NOT NULL REFERENCES Venue(VenID)
);
CREATE TABLE WorksAt (
    StaffID INTEGER REFERENCES Staff(StaffID),
    ConID INTEGER REFERENCES Concert(ConID),
    PRIMARY KEY (StaffID, ConID)
);
```

Information
Flag question

Consider the following relation:

- Sales (ProductID, ProductName, Date, Quantity, Revenue)

Question 29
Complete
Marked out of 2.00
Flag question

Consider the following query on the Sales relation:

UPDATE Sales

SET Revenue = Revenue * 1.1

WHERE ProductID = 12345 AND Date = '2023-03-10';

Select the true statements:

- ☐ a. This is an example of an OLAP query.
- ☑ b. This is an example of an OLTP query.
- ☐ c. Using a column store storage layout for this query is better than using a row store.

Question 30
Complete
Marked out of 2.00
Flag question

Consider the following query on the Sales relation:

SELECT ProductID, SUM(Quantity) AS TotalQuantitySold

FROM Sales

WHERE Date BETWEEN '2023-01-01' AND '2023-12-31'

GROUP BY ProductID;

Select the true statements:

- ☑ a. Using a column store storage layout for this query is better than using a row store.
- ☐ b. This is an example of an OLTP query.
- ☑ c. This is an example of an OLAP query.

Question 31
Complete
Marked out of 1.00
Flag question

Consider the following query on the Sales relation:

SELECT *

FROM Sales S1

JOIN Sales S2 ON S1.ProductID > S2.ProductID

WHERE S1.ProductName = S2.ProductName;

Which join algorithm would an optimizer choose for the above query?

- ○ a. Nested Loop Join
- ○ b. Hash Join
- ● c. Merge Join

Question 32
Complete
Marked out of 5.00
Flag question

Select the true statements:

- ☐ a. 'C' stands for Consistency in ACID and the CAP theorem and has the same meaning in both cases.
- ☑ b. Spark can process unstructured data.
- ☐ c. Relational databases are superior to NoSQL databases in all use cases.
- ☐ d. Big data refers to the complexity of the data, not its size.
- ☑ e. Key-value stores are not suitable for applications that involve frequent complex joins.
- ☐ f. Sequential disk writes are the most important disk access pattern in big data analytics.

Information
Flag question

Suppose that the database of a university contains, among others, the following relations (all containing a considerable number of tuples):

- Students(id, name, street, city, immatriculation_year)
- Lectures(id, name, credits, semester)
- Attendance(lectures_id, students_id)

Select the indexes (could be 0, 1, or more) that a query optimizer would use to process each of the queries below. Assume a data distribution and selectivity such that, whenever a relevant index is available, an index-scan is better than a full scan. Indicate whether one of the selected indexes is covering by checking the "It is covering" box.

Question 33
Complete
Marked out of 2.00
Flag question

What are the best index(es) for the query below:

SELECT DISTINCT city, street

FROM Students;

- ☐ a. Students(id,city)
- ☑ b. Students(city)
- ☐ c. It is covering
- ☐ d. Students(id)
- ☐ e. Students(city, id)
- ☐ f. No index

Question 34
Complete
Marked out of 2.50
Flag question

What are the best index(es) for the query below:

SELECT id

FROM lectures

WHERE credits = (SELECT min(credits) FROM lectures);

- ☐ a. Lectures(id, credits)
- ☑ b. It is covering
- ☐ c. No index
- ☐ d. Lectures(id)
- ☑ e. Lectures(credits,id)
- ☐ f. Lectures(credits,id,name)

Question 35
Complete
Marked out of 2.50
Flag question

What are the best index(es) for the query below:

SELECT name, city

FROM Students

WHERE immatriculation_year = '2022';

- ☐ a. It is covering
- ☐ b. Students(city)
- ☐ c. Students(name)
- ☑ d. Students(immatriculation_year)
- ☐ e. No index
- ☐ f. Students(city, street)

Question 36
Complete
Marked out of 3.00
Flag question

What are the best index(es) for the query below:

SELECT Lectures.name, Lectures.semester

FROM Attendance

JOIN Lectures ON Attendance.lectures_id = Lectures.id

WHERE Attendance.students_id = 123456;

- ☑ a. No index
- ☐ b. It is covering
- ☐ c. Lectures(name)
- ☐ d. Lectures(id)
- ☐ e. Lectures(id,name)
- ☐ f. Attendance(students_id)

Finish review