WGU Data-Management-Foundations D426 Practice Questions

Shared by Page on 30-10-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which operation finds an entry containing a search value by repeatedly splitting the index in two?

Options:

A- Table scan
B- Binary search
C- Index scan
D- Fan-out

P2P exams

Answer:

B

Explanation:

Binary search is an algorithm that finds a search value by repeatedly dividing the search space into two halves. It is commonly used in indexed searches.

Example Usage in Databases:

B-Trees and B+ Trees (used in database indexing) apply binary search to navigate the index efficiently.

If searching for ID = 50 in a sorted list of IDs, binary search:

Splits the list into two halves.

Checks the middle value.

Eliminates half of the dataset.

Repeats until the value is found.

Why Other Options Are Incorrect:

Option A (Table scan) (Incorrect): Reads every row, much slower than binary search.

Option C (Index scan) (Incorrect): Uses indexes but does not necessarily apply binary search.

Option D (Fan-out) (Incorrect): Describes branching in B-Trees, not searching.

Thus, the correct answer is Binary search, as it repeatedly splits the index in two.

Question 2

Question Type: MultipleChoice

Which phase of entity-relationship modeling refers to the maxima and minima of relationships and attributes?

Options:

A- Cardinality
B- Physical design
C- Attribute minimum
D- Partition

Answer:

A

Explanation:

Cardinality defines the minimum and maximum number of occurrences of one entity in relation to another.

Example Usage in an ER Model:

One-to-Many (1:M): A customer can place multiple orders, but each order belongs to only one customer.

Customers (1) --------- (M) Orders

Cardinality notation:

(1,1) One-to-One

(0,M) Zero-to-Many

(1,M) One-to-Many

Why Other Options Are Incorrect:

Option B (Physical design) (Incorrect): Focuses on storage and indexing, not relationships.

Option C (Attribute minimum) (Incorrect): No such formal term in database modeling.

Option D (Partition) (Incorrect): Refers to dividing tables, not relationship constraints.

Thus, the correct answer is Cardinality, as it defines min/max constraints on relationships.

Question 3

Question Type: MultipleChoice

What is the role of the database administrator?

Options:

A- The database administrator is a consumer of data in a database.
B- The database administrator is responsible for securing the database system against unauthorized users.
C- The database administrator determines the format of each data element and the overall database structure.
D- The database administrator develops computer programs that utilize a database.

Answer:

B

Explanation:

A Database Administrator (DBA) is responsible for the management, security, and performance of a database system. This includes controlling access to data, ensuring database integrity, optimizing performance, managing backups, and protecting the system from unauthorized access.

Option A (Incorrect): A DBA is not just a consumer of data but is primarily responsible for the database's management.

Option B (Correct): Security is one of the key responsibilities of a DBA, including enforcing user access controls and implementing encryption and authentication mechanisms.

Option C (Incorrect): While DBAs work with data structures, it is typically the role of a data architect or database designer to define data formats and schema structures.

Option D (Incorrect): Developing application programs that interact with the database is typically the role of software developers or database programmers, not DBAs.

Question 4

Question Type: MultipleChoice

How can a main key constraint be added after the table is created?

Options:

A- By using an ALTER clause
B- By using the CREATE TABLE statement
C- By using an UPDATE clause
D- By using an INSERT INTO clause

Answer:

A

Explanation:

To add a primary key constraint after table creation, we use the ALTER TABLE statement.

Example Usage:

sql

ALTER TABLE Employees

ADD CONSTRAINT PK_Employees PRIMARY KEY (EmpID);

This adds a primary key to the EmpID column after the table was created.

Why Other Options Are Incorrect:

Option B (CREATE TABLE) (Incorrect): Used for defining constraints during table creation, not after.

Option C (UPDATE) (Incorrect): Modifies row values, not constraints.

Option D (INSERT INTO) (Incorrect): Used to add data to a table, not modify constraints.

Thus, the correct answer is ALTER TABLE, as it modifies table structure to add a primary key constraint.

Question 5

Question Type: MultipleChoice

Where does a primary key traditionally appear in a table?

Options:

A- In the table header
B- In the first column
C- In the top row
D- In the last visible column

P2P exams

Answer:

B

Explanation:

By database design conventions, the primary key is usually placed in the first column of a table to make it easy to identify and reference.

Example Usage:

sql

CREATE TABLE Employees (

EmpID INT PRIMARY KEY, -- First column (convention)

Name VARCHAR(50),

Salary DECIMAL(10,2)

);

P2P exams

EmpID is placed as the first column for clarity and quick access.

Why Other Options Are Incorrect:

Option A (In the table header) (Incorrect): Table headers only display column names, they do not contain values.

Option C (In the top row) (Incorrect): The top row contains data, not the primary key definition.

Option D (In the last visible column) (Incorrect): While technically possible, placing a primary key at the end is uncommon in database design.

Thus, the correct answer is In the first column, as this is the standard convention in relational databases.

Question 6

Question Type: MultipleChoice

Which property of an entity can become a column in a table?

Options:

A- Modality
B- Uniqueness
C- Attribute
D- Non-null values

Answer:

C

Explanation:

In database design, attributes of an entity become columns in a relational table.

Example Usage:

For an Employee entity, attributes might include:

Entity	Attributes (Columns in Table)
Employee	`EmployeeID`, `Name`, `Salary`, `DepartmentID`

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

Name VARCHAR(50),

Salary DECIMAL(10,2),

DepartmentID INT

);

Each attribute (e.g., Name, Salary) becomes a column in the table.

Why Other Options Are Incorrect:

Option A (Modality) (Incorrect): Describes optional vs. mandatory relationships, not table structure.

Option B (Uniqueness) (Incorrect): Ensures distinct values but is not a column property.

Option D (Non-null values) (Incorrect): Ensures that columns must contain data but does not define attributes.

Thus, the correct answer is Attribute, as attributes of entities become table columns.

Question 7

Question Type: MultipleChoice

Which operator defines the field that the index is using in a CREATE TABLE statement?

Options:

A- ON
B- IN
C- UNIQUE
D- CHECK

Answer:

A

Explanation:

The ON keyword specifies the field used by an index when creating it in SQL.

Example Usage:

sql

CREATE INDEX idx_employee_name

ON Employees(Name);

Here, an index idx_employee_name is created on the Name column.

This improves query performance when filtering by Name.

Why Other Options Are Incorrect:

Option B (IN) (Incorrect): Used in queries to match values in a set, not for indexing.

Option C (UNIQUE) (Incorrect): Ensures a column has unique values but does not define an index field.

Option D (CHECK) (Incorrect): Used for validating column values, not for indexing.

Thus, the correct answer is ON, as it defines the column on which an index is created.

Question 8

Question Type: MultipleChoice

Which command is used to filter group results generated by the GROUP BY clause?

Options:

A- REPLACE
B- HAVING
C- WITH
D- WHERE

Answer:

B

Explanation:

The HAVING clause is used in SQL to filter grouped results generated by the GROUP BY clause. Unlike WHERE, which filters individual rows before grouping, HAVING filters after aggregation has been performed.

Example Usage:

sql

SELECT Department, AVG(Salary) AS AvgSalary

FROM Employees

GROUP BY Department

HAVING AVG(Salary) > 50000;

This query first groups employees by Department, calculates the average salary per department, and then filters only those departments where the average salary is greater than 50,000.

Why Other Options Are Incorrect:

Option A (REPLACE) (Incorrect): Used for string substitution, not filtering.

Option C (WITH) (Incorrect): Used in Common Table Expressions (CTEs), not for filtering.

Option D (WHERE) (Incorrect): Used for row-level filtering before aggregation, but it cannot be used on aggregate functions like SUM() or AVG().

Thus, HAVING is the correct answer for filtering after grouping.

Question 9

Question Type: MultipleChoice

Which term refers to a path from a top-level block to a bottom-level block?

Options:

A- Fan-out
B- Crow's foot
C- Branch
D- Sparse index

Answer:

C

Explanation:

In database indexing, a branch refers to the path from the top-level block (root node) to a bottom-level block (leaf node) in a B-Tree or B+ Tree index structure.

Example Usage in Indexing:

A B-Tree index organizes data hierarchically, with branches leading to different parts of the tree.

When searching for a record, the query follows a branch from the root node down to the correct leaf node.

Why Other Options Are Incorrect:

Option A (Fan-out) (Incorrect): Refers to how many children a node has, not the path.

Option B (Crow's foot) (Incorrect): A notation used in ER diagrams, not indexing.

Option D (Sparse index) (Incorrect): A type of index storing only some entries, not the path itself.

Thus, the correct answer is Branch, as it defines the path from top to bottom in a database index.

P2P exams

Question 10

Question Type: MultipleChoice

Which main characteristic is used to differentiate between strong and weak entities in the process of database design?

Options:

A- Association
B- Primary key
C- Foreign key
D- Cardinality

P2P exams

Answer:

B

Explanation:

In database design, an entity is classified as strong or weak based on whether it has a primary key that uniquely identifies its records.

Differences Between Strong and Weak Entities:

Entity Type	Primary Key	Example
Strong Entity	Has its own unique primary key	`Student (StudentID)`
Weak Entity	Does not have a primary key of its own; depends on another table's key	`OrderDetails (Depends on OrderID)`

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT

);

CREATE TABLE OrderDetails (

OrderDetailID INT,

OrderID INT,

ProductID INT,

PRIMARY KEY (OrderDetailID, OrderID),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)

);

Orders is a strong entity (has OrderID as its own primary key).

OrderDetails is a weak entity (depends on OrderID for uniqueness).

Why Other Options Are Incorrect:

Option A (Association) (Incorrect): Associations describe relationships but do not define strong/weak entities.

Option C (Foreign key) (Incorrect): Weak entities depend on foreign keys, but primary keys define their status.

Option D (Cardinality) (Incorrect): Cardinality defines relationship constraints but does not differentiate entity types.

Thus, the correct answer is Primary key, as it is the defining characteristic between strong and weak entities.

To Get Premium Files for Data-Management-Foundations Visit

https://www.p2pexams.com/products/data-management-foundations

For More Free Questions Visit

https://www.p2pexams.com/wgu/pdf/data-management-foundations

