**D427 Practice Test 2**

1. Portland, OR 97212

How many attributes are present in the address fragment?

1. 1
2. 2
3. 3
4. 4
5. The Patient table has the following columns:

first\_name – varchar(20)

last\_name – varchar(30)

birthdate – date

patient\_id – int

Which column should be designated at the primary key for the Patient table?

1. first\_name
2. last\_name
3. birthdate
4. patient\_id
5. The Exam table has the following columns:

exam\_id – int

exam\_date – date

exam\_reason – varchar (100)

patient\_id – int

Which column should be designated as the foreign key for the Exam table?

1. exam\_id
2. exam\_date
3. exam\_reason
4. patient\_id
5. Which data type represents numbers with fractional values:
6. Varchar
7. Integer
8. Binary
9. Decimal
10. Which of the following is a DDL (Data Definition Language) command?
11. INSERT
12. SELECT
13. CREATE INDEX
14. UPDATE
15. Which of the following is a DML(Data Manipulation Language) command?
16. CREATE VIEW
17. CREATE TABLE
18. INSERT
19. ALTER INDEX

CREATE TABLE Exam (

exam\_id INT NOT NULL AUTO\_INCREMENT,

exam\_date DATE NOT NULL,

exam\_reason VARCHAR (100),

patient\_id INT NOT NULL,

PRIMARY KEY (exam\_id),

FOREIGN KEY (patient\_id) REFERENCES Patient (patient \_id) **ON DELETE CASCADE**

);

A picture containing text, screenshot, font, number

Description automatically generated

Looking at the Patient and Exam tables above and the CREATE TABLE for the Exam table with foreign key reference statement above, what would happen to exams in the Exam table that are linked to a patient if that patient is deleted.

1. Those exams would remain in the database.
2. Those exams would be deleted also.
3. The Patient ID for those exams would be changed to NULL.
4. Nothing would happen.

CREATE TABLE Exam (

exam\_id INT NOT NULL AUTO\_INCREMENT,

exam\_date DATE NOT NULL,

exam\_reason VARCHAR (100),

patient\_id INT NOT NULL,

PRIMARY KEY (exam\_id),

FOREIGN KEY (patient\_id) REFERENCES Patient (patient \_id) **ON DELETE RESTRICT**

);

A picture containing text, screenshot, font, number

Description automatically generated

Looking at the Patient and Exam tables above and the CREATE TABLE for the Exam table with foreign key reference statement above, what would happen to exams in the Exam table that are linked to a patient if that patient is deleted.

1. Those invoices would remain in the database.
2. Those invoices would be deleted also.
3. The Customer ID for those invoices would be changed to NULL.
4. The delete of the Customer would not be allowed.

CREATE TABLE Exam (

exam\_id INT NOT NULL AUTO\_INCREMENT,

exam\_date DATE NOT NULL,

exam\_reason VARCHAR (100),

patient\_id INT NOT NULL,

PRIMARY KEY (exam\_id),

FOREIGN KEY (patient\_id) REFERENCES Patient (patient \_id) **ON DELETE SET TO NULL**

);

A picture containing text, screenshot, font, number

Description automatically generated

Looking at the Patient and Exam tables above and the CREATE TABLE for the Exam table with foreign key reference statement above, what would happen to exams in the Exam table that are linked to a patient if that patient is deleted.

1. Those invoices would remain in the database.
2. Those invoices would be deleted also.
3. The Customer ID for those invoices would be changed to NULL.
4. The delete of the Customer would not be allowed.

10.

Which of the following are true about materialized view (Choose 2)?

1. It is a base table.
2. It is stored.
3. It must be refreshed whenever the base table changes.
4. The results are stored as a temporary table.

11.

The Student table will have the following columns:  
StudentID—positive integer  
FirstName—variable-length string with up to 50 characters  
MiddleInitial—fixed-length string with 1 character  
LastName—variable-length string with up to 50 characters  
DateOfBirth—date  
AccountBalance—positive decimal value representing a balance of up to $24,999, with 2 digits for cents

Write a SQL statement to create the Student table.  
Do not add any additional constraints to any column beyond what is stated.

12.

The Classification table has the following columns:  
ClassificationCode—integer, primary key  
ClassificationDescription—variable-length string

The Vehicle table should have the following columns:  
Name—variable-length string, maximum 30 characters  
ClassificationCode—integer

Write a SQL statement to create the Vehicle table. Designate the ClassificationCode column in the Vehicle table as a foreign key to the ClassificationCode column in the Classification table.

13.

The Vehicle table has the following columns:

ID—integer, primary key  
Make—variable-length string  
Model—variable-length string  
Year—integer

A new column must be added to the Automobile table:  
Column name: EngineSize  
Data type: decimal (2,1)

Write a SQL statement to add the EngineSize column to the Vehicle table.

14.

The Song table has the following columns:  
ID—integer, primary key  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL statement to create a view named MyMusic that contains the Title, Genre, and Year columns for all ~~movies~~ books. Ensure your result set returns the columns in the order indicated.

15.

A database has a view named MyMusic.  
Write a SQL statement to delete the view named MyMusic from the database.

16.

The Pet table has the following columns:  
petID - integer, primary key  
name - variable-length string  
breed - variable-length string  
birthdate - date

Write a SQL statement to modify the Pet table to make the petID column the primary key.

17.

The Dog table has the following columns:  
dogID - integer, primary key  
name - variable-length string  
breedID - integer  
birthdate - date

The Breed table has the following columns:  
breedID—integer  
breedDescription—varchar  
Releases—integer

Write a SQL statement to designate the breedID column in the Dog table as a foreign key to the breedID column in the Breed table.

18.

The Song table has the following columns:  
ID—integer, primary key  
Title—variable-length string  
Genre—variable-length string  
Year—integer  
Write a SQL statement to create an index named idx\_year on the Year column of the Song table.

19.

The Podcast table has the following columns:  
podcastID—integer, primary key, auto\_increment  
Title—variable-length string  
Speaker—variable-length string  
Minutes—integer

The following data needs to be added to the Podcast table:  
Title Speaker Minutes  
Rock Painting, Kecia McDonald, 25

Write a SQL statement to insert the indicated data into the Podcast table.

20.

The Podcast table has the following columns:  
podcastID—integer, primary key, auto\_increment  
Title—variable-length string  
Speaker—variable-length string  
Minutes—integer

Write a SQL statement to delete the row with the ID value of 33 from the Podcast table.

21.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL statement to update the Year value to be 2022 for all books with a Year value of 2020.

22. Which query illustrates performing an outer join of the Movie table with a different table?

1. SELECT M.Title, A.Name FROM Movie M, Actor A  
   WHERE M.ActorID = A.ActorID;
2. SELECT M.Title, A.Name FROM Movie M, Actor A  
   WHERE M.ActorID = A.MovieID;
3. SELECT M.Title, A.Name FROM Movie M RIGHT JOIN Actor A  
   ON M.ActorID = A.ActorID;
4. SELECT M.Title, A.Actor FROM Movie M  
   INNER JOIN Actor A ON M.ActorID = A.ActorID

23.

Assume there are two tables, A and B.  
Which rows will always be included in the result set if Table A is inner joined with Table B?

a. Only rows in Tables A and B that share the join condition

b. All rows in Table B

c. All rows in Table A

d. Only rows in Tables A and B that do not share the join condition.

24.

The database contains a table named Book.  
Write a SQL query to return all data from the Book table without directly referencing any column names.

25.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL query to retrieve the Title and Genre values for all records in the Book table with a Year value of 2020. Ensure your result set returns the columns in the order indicated.

26.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL query to display all Title values in alphabetical order A–Z.

27.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL query to output the unique Genre values and the number of books with each genre value from the Book table as GenreCount. Sort the results by the Genre in alphabetical order A–Z. Ensure your result set returns the columns in the order indicated.

28.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

The YearSales table has the following columns:  
Year—integer  
TotalSales—bigint unsigned  
Releases—integer  
Write a SQL query to display both the Title and the TotalSales (if available) for all books. Ensure your result set returns the columns in the order indicated.

29.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL query to return how many books have a Year value of 2019.

30. A(n) \_\_\_\_\_ is a query that is embedded (or nested) inside another query.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a. | alias | b. | operator |
|  | c. | subquery | d. | view |

31. All changes to a table structure are made using the \_\_\_\_\_ command, followed by a keyword that produces the specific changes a user wants to make.

1. ALTER TABLE
2. UPDATE TABLE
3. COMMIT TABLE
4. DELETE TABLE

32. The SQL aggregate function that gives the number of rows containing non-null values for a given column is \_\_\_\_\_.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a. | COUNT | b. | MIN |
|  | c. | MAX | d. | SUM |

|  |  |  |
| --- | --- | --- |
| 33. The \_\_\_\_\_ condition of a JOIN is generally composed of an equality comparison between the foreign key and the primary key of related tables.  34. Which kind of relationship is displayed in the entity-relationship diagram below?   1. Binary one-to-one 2. Unary many-to-many 3. Ternary one-to-one 4. Binary one-to-many 5. Unary one-to-one  |  |  | | --- | --- | |  |  | |
| A picture containing text, screenshot, font, number  Description automatically generated |

35. Which kind of relationship is displayed in the entity-relationship diagram below?

1. Binary one-to-one
2. Unary many-to-many
3. Ternary one-to-one
4. Binary many-to-many
5. Unary one-to-one

A picture containing text, screenshot, font, number

Description automatically generated

36. Which kind of relationship is displayed in the entity-relationship diagram below?

1. Binary one-to-one
2. Unary many-to-many
3. Ternary one-to-one
4. Binary many-to-many
5. Unary one-to-one

A diagram of a work flow

Description automatically generated with low confidence

37. Which data type can be designated to allow for storage of dates

1. VARCHAR
2. INTEGER
3. DECIMAL
4. TIMESTAMP/DATE

38.

A picture containing text, font, number, screenshot

Description automatically generated

Which query would produce a result set that listed all of the patients, regardless of whether they had an appointment in the Exam table or not.

1. SELECT patient\_id, first\_name, last\_name, birthdate, exam\_date

FROM Patient RIGHT JOIN EXAM

ON Patient.patient\_id = Exam.patient\_id;

1. SELECT patient\_id, first\_name, last\_name, birthdate, exam\_date

FROM Patient INNER JOIN EXAM

ON Patient.patient\_id = Exam.patient\_id;

1. SELECT patient\_id, first\_name, last\_name, birthdate, exam\_date

FROM Patient INNERJOIN EXAM

ON Patient.patient\_id = Exam.exam\_id;

1. SELECT patient\_id, first\_name, last\_name, birthdate, exam\_date

FROM Patient LEFT JOIN EXAM

ON Patient.patient\_id = Exam.patient\_id;

39.

A picture containing text, screenshot, font, businesscard

Description automatically generated

Which query would produce a result set that listed all of the Model Types, regardless of whether a model was assigned to that type.

1. SELECT model\_id, lname, Models.model\_type\_id, hourly\_fee

FROM Models

RIGHT JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;

1. SELECT model\_id, lname, Models.model\_type\_id, hourly\_fee

FROM Models

LEFT JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;

1. SELECT model\_id, lname, Models.model\_type\_id, hourly\_fee

FROM Models

INNER JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;

1. SELECT model\_id, lname, Models.model\_type\_id, hourly\_fee

FROM Models

NATURAL JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;

40.

Write a query to list Models along with the model type they are assigned. Include all model information and the hourly fee.

41.

Write a query to delete Amy Lin from the Customer table shown below.

A picture containing text, number, screenshot, font

Description automatically generated

42. Write a SQL statement to update Blanca Garcia’s phone number in the Customer table below to 555-222-1234.

A picture containing text, number, screenshot, font

Description automatically generated

43. Write a SQL statement to retrieve all of the Customers from Seattle from the table below.

A picture containing text, number, screenshot, font

Description automatically generated

44. Write the CREATE TABLE statement to create the Exam table below. Make sure to designate the primary key and foreign key of the table.

A picture containing text, screenshot, font, number

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Bagel ID** | **Bagel Name** | **Bagel Description** | **Cost/Bagel** |
| RB | Radically Raisin Bagel | Raisin Bagel | 2.00 |
| EB | All the Stuff Bagel | Everything Bagel | 2.35 |
| CB | Chip off the Block Bagel | Chocolate Chip Bagel | 2.25 |
| MB | Keepin’ it Healthy Bagel | Multigrain Bagel | 2.00 |
| NB | Classic New York Bagel | New York Bagel | 1.75 |

45. Write a query to pull all of the data from the Bagel table below. Order the result set alphabetically by Bagel ID.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer ID** | **First Name** | **Last Name** | **Address** | **City** | **State** | **Zip** |
| 101 | Amy | Lin | 123 A Street | Seattle | WA | 98118 |
| 102 | Viya | Prasad | 111 B Avenue | Seattle | WA | 98118 |
| 104 | Blanca | Garcia | 8674 Main Street | Portland | OR | 97212 |
| 107 | Shelia | Brown | 22815 Foster Road | Portland | OR | 97217 |
| 108 | Anna | Cruz | 444 Apple Lane | Troutdale | OR | 97708 |
| 110 | Amy | Smith | 39878 Acorn Place | Chicago | IL | 60435 |

46. Write a query to count the number of customers in each city.

**Customer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer ID** | **First Name** | **Last Name** | **Address** | **City** | **State** | **Zip** |
| 101 | Amy | Lin | 123 A Street | Seattle | WA | 98118 |
| 102 | Viya | Prasad | 111 B Avenue | Seattle | WA | 98118 |
| 104 | Blanca | Garcia | 8674 Main Street | Portland | OR | 97212 |
| 107 | Shelia | Brown | 22815 Foster Road | Portland | OR | 97217 |
| 108 | Anna | Cruz | 444 Apple Lane | Troutdale | OR | 97708 |
| 110 | Amy | Smith | 39878 Acorn Place | Chicago | IL | 60435 |

47. Write a query that will return all of the cities from the Customer table above. Do not include duplicates -list each city only once.

48. Write a query to list customers from Chicago or Seattle.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer ID** | **First Name** | **Last Name** | **Address** | **City** | **State** | **Zip** |
| 101 | Amy | Lin | 123 A Street | Seattle | WA | 98118 |
| 102 | Viya | Prasad | 111 B Avenue | Seattle | WA | 98118 |
| 104 | Blanca | Garcia | 8674 Main Street | Portland | OR | 97212 |
| 107 | Shelia | Brown | 22815 Foster Road | Portland | OR | 97217 |
| 108 | Anna | Cruz | 444 Apple Lane | Troutdale | OR | 97708 |
| 110 | Amy | Smith | 39878 Acorn Place | Chicago | IL | 60435 |

49. Write a join statement that pulls the values listed from 3 of the tables below (Volunteer, Registration, and Activity Registration).

VolunteerID

LastName

Registration Date

ActivityID

A picture containing text, screenshot, font, number

Description automatically generated

50. Write a join statement that pulls the values listed from 3 of the tables below (Customer, Invoice, and Invoice Item).

CustomerID

InvoiceID

Date

ProductID

A picture containing text, screenshot, font, line

Description automatically generated

51. Write a join statement that pulls the values listed from 3 of the tables below (Models, Model Types, Phone Numbers).

model\_id

last\_name  
first\_name

model\_type\_id

phone\_number

A picture containing text, screenshot, font, number

Description automatically generated

52.

**student\_grades**

|  |  |  |
| --- | --- | --- |
| studentID | lastName | grade |
| 44444 | Jones | B |
| 24352 | Chang | A |
| 98724 | Wu | B |
| 98789 | Chatham | A |
| 98887 | Fredricks | C |
| 00878 | Moore | A |
| 22078 | Prasad | C |
| 79692 | Perez | A |

Refer to the given SELECT statement and the table above.  
  
SELECT grade, COUNT(\*) AS gradeCount   
FROM student\_grades  
GROUP BY grade;

Which clause added to the statement finds grades with a count greater than 2?

a. WITH gradeCount <> DUPLICATE

b. NOT gradeCount = NULL

c. HAVING gradeCount > 2

d. INCLUDING gradeCount > 2

53. Using the student\_grades table listed above, write a query to count how many students have each grade (A, B, and C).

A picture containing text, screenshot, font, number

Description automatically generated

54. Using the Model\_Types table in the E-R diagram above, write a query to find the highest model fee.

55. Write a query to update the hourly\_fee to 100.00 more for model types with an hourly\_fee less that 600.00.

56.   
Which query illustrates performing an outer equijoin of the Movie table with a different table?

a. SELECT M. Title, A.Actor FROM Movie M, Actor A  
 WHERE M.ActorID = [A.ID](https://protect-us.mimecast.com/s/mEb1CkRjpZCYzJg0yu27JuB?domain=a.id)  
  
b. SELECT M.Title, A.Actor FROM Movie M  
 LEFT JOIN Movie MB ON [M.ID](https://protect-us.mimecast.com/s/V00QClYkq5cONYLKwT9fSw6?domain=m.id) = [MB.ID](https://protect-us.mimecast.com/s/H5f1CmZ0rQU1NQvYohBEwTN?domain=mb.id), Actor A  
  
c. SELECT M. Title, A.Actor FROM Movie M  
 RIGHT JOIN Actor A ON M.ActorID = [A.ID](https://protect-us.mimecast.com/s/mEb1CkRjpZCYzJg0yu27JuB?domain=a.id)  
  
d. SELECT M. Title, A.Actor FROM Movie M  
 INNER JOIN Actor A ON M. ActorID = [A.ID](https://protect-us.mimecast.com/s/mEb1CkRjpZCYzJg0yu27JuB?domain=a.id)

57. Which rows will always be included in the result set if Table A is right joined with Table B?

1. Only rows in Tables A and B that share the join condition
2. All rows in Table B
3. All rows in Table A
4. Only rows in Tables A and B that do not share the join condition

58.

The Dog table has the following columns:

CREATE TABLE dog (  
dogID INT NOT NULL,  
name VARCHAR(10) NOT NULL,  
breedID INT,  
birthdate DATE

);

Modify the name column so that the variable length string has a maximum of 25 characters.

**D427 Practice Test 2 – ANSWER KEY**

1. Portland, OR 97212

How many attributes are present in the address fragment?

1. 1
2. 2
3. 3
4. 4
5. The Patient table has the following columns:

first\_name – varchar(20)

last\_name – varchar(30)

birthdate – date

patient\_id – int

Which column should be designated at the primary key for the Patient table?

1. first\_name
2. last\_name
3. birthdate
4. patient\_id
5. The Exam table has the following columns:

exam\_id – int

exam\_date – date

exam\_reason – varchar (100)

patient\_number – int

Which column should be designated as the foreign key for the Exam table?

1. exam\_id
2. exam\_date
3. exam\_reason
4. patient\_id
5. Which data type represents numbers with fractional values:
6. Varchar
7. Integer
8. Binary
9. Decimal
10. Which of the following is a DDL command?
11. INSERT
12. SELECT
13. CREATE INDEX
14. UPDATE
15. Which of the following is a DML command?
16. CREATE VIEW
17. CREATE TABLE
18. INSERT
19. ALTER INDEX

CREATE TABLE Exam (

exam\_id INT NOT NULL AUTO\_INCREMENT,

exam\_date DATE NOT NULL,

exam\_reason VARCHAR (100),

patient\_id INT NOT NULL,

PRIMARY KEY (exam\_id),

FOREIGN KEY (patient\_id) REFERENCES Patient (patient \_id) **ON DELETE CASCADE**

);

A picture containing text, screenshot, font, number

Description automatically generated

Looking at the Patient and Exam tables above and the CREATE TABLE for the Exam table with foreign key reference statement above, what would happen to exams in the Exam table that are linked to a patient if that patient is deleted.

1. Those exams would remain in the database.
2. Those exams would be deleted also.
3. The Patient ID for those exams would be changed to NULL.
4. Nothing would happen.

CREATE TABLE Exam (

exam\_id INT NOT NULL AUTO\_INCREMENT,

exam\_date DATE NOT NULL,

exam\_reason VARCHAR (100),

patient\_id INT NOT NULL,

PRIMARY KEY (exam\_id),

FOREIGN KEY (patient\_id) REFERENCES Patient (patient \_id) **ON DELETE RESTRICT**

);

A picture containing text, screenshot, font, number

Description automatically generated

Looking at the Patient and Exam tables above and the CREATE TABLE for the Exam table with foreign key reference statement above, what would happen to exams in the Exam table that are linked to a patient if that patient is deleted.

1. Those invoices would remain in the database.
2. Those invoices would be deleted also.
3. The Customer ID for those invoices would be changed to NULL.
4. The delete of the Customer would not be allowed.

CREATE TABLE Exam (

exam\_id INT NOT NULL AUTO\_INCREMENT,

exam\_date DATE NOT NULL,

exam\_reason VARCHAR (100),

patient\_id INT NOT NULL,

PRIMARY KEY (exam\_id),

FOREIGN KEY (patient\_id) REFERENCES Patient (patient \_id) **ON DELETE SET TO NULL**

);

A picture containing text, screenshot, font, number

Description automatically generated

Looking at the Patient and Exam tables above and the CREATE TABLE for the Exam table with foreign key reference statement above, what would happen to exams in the Exam table that are linked to a patient if that patient is deleted.

1. Those invoices would remain in the database.
2. Those invoices would be deleted also.
3. The Customer ID for those invoices would be changed to NULL.
4. The delete of the Customer would not be allowed.

10.

Which of the following are true about materialized view (Choose 2)?

1. It is a base table.
2. It is stored.
3. It must be refreshed whenever the base table changes.
4. The results are stored as a temporary table.

11.

The Student table will have the following columns:  
StudentID—positive integer  
FirstName—variable-length string with up to 50 characters  
MiddleInitial—fixed-length string with 1 character  
LastName—variable-length string with up to 50 characters  
DateOfBirth—date  
AccountBalance—positive decimal value representing a balance of up to $24,999, with 2 digits for cents

Write a SQL statement to create the Student table.  
Do not add any additional constraints to any column beyond what is stated.

**CREATE TABLE Student (**

**StudentID INT UNSIGNED,**

**FirstName VARCHAR(50),**

**MiddleInitial CHAR(1),**

**LastName VARCHAR(50),**

**DateOfBirth DATE,**

**AccountBalance DECIMAL(7,2) UNSIGNED**

);

12.

The Classification table has the following columns:  
ClassificationCode—integer, primary key  
ClassificationDescription—variable-length string

The Vehicle table should have the following columns:  
Name—variable-length string, maximum 30 characters  
ClassificationCode—integer

Write a SQL statement to create the Vehicle table. Designate the ClassificationCode column in the Vehicle table as a foreign key to the ClassificationCode  column in the Classification table.

**CREATE TABLE Vehicle (**

**Name VARCHAR(30),**

**ClassificationCode INT,**

**FOREIGN KEY (ClassificationCode) REFERENCES Classification(ClassificationCode)**

**);**

13.

The Vehicle table has the following columns:

ID—integer, primary key  
Make—variable-length string  
Model—variable-length string  
Year—integer

A new column must be added to the Vehicle table:  
Column name: EngineSize  
Data type: decimal (2,1)

Write a SQL statement to add the EngineSize column to the Vehicle table.

**ALTER TABLE Vehicle**

**ADD** EngineSize **DECIMAL(2,1);**

14.

The Song table has the following columns:  
ID—integer, primary key  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL statement to create a view named MyMusic that contains the Title, Genre, and Year columns for all ~~movies~~ books. Ensure your result set returns the columns in the order indicated.

**CREATE VIEW MyMusic AS**

**SELECT Title, Genre, Year**

**FROM Song;**

15.

A database has a view named MyMusic.  
Write a SQL statement to delete the view named MyMusic from the database.

**DROP VIEW MyMusic;**

16.

The Pet table has the following columns:  
petID - integer, primary key  
name - variable-length string  
breed - variable-length string  
birthdate - date

Write a SQL statement to modify the Pet table to make the petID column the primary key.

**ALTER TABLE Pet**

**ADD PRIMARY KEY (petID);**

17.

The Dog table has the following columns:  
dogID - integer, primary key  
name - variable-length string  
breedID - integer  
birthdate - date

The Breed table has the following columns:  
breedID—integer  
breedDescription—varchar  
Releases—integer

Write a SQL statement to designate the breedID column in the Dog table as a foreign key to the breedID column in the Breed table.

**ALTER TABLE Dog**

**ADD FOREIGN KEY (breedID) REFERENCES Breed (breedID);**

18.

The Song table has the following columns:  
ID—integer, primary key  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL statement to create an index named idx\_year on the Year column of the Song table.

**CREATE INDEX idx\_year ON Song (Year);**

19.

The Podcast table has the following columns:  
podcastID—integer, primary key, auto\_increment  
Title—variable-length string  
Speaker—variable-length string  
Minutes—integer

The following data needs to be added to the Podcast table:  
Title Speaker URL  
Rock Painting, Kecia McDonald, 25

Write a SQL statement to insert the indicated data into the Podcast table.

**INSERT INTO Podcast (Title, Speaker, Minutes) VALUES**

**(‘Rock Painting’, ‘Kecia McDonald’, 25);**

20.

The Podcast table has the following columns:  
podcastID—integer, primary key, auto\_increment  
Title—variable-length string  
Speaker—variable-length string  
Minutes—integer

Write a SQL statement to delete the row with the ID value of 33 from the Podcast table.

**DELETE from Podcast WHERE ID = 33;**

21.

The Song table has the following columns:  
ID—integer, primary key  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL statement to update the Year value to be 2022 for all books with a Year value of 2020.

**UPDATE Song  
SET Year = 2022  
WHERE Year = 2020;**

22. Which query illustrates performing an outer join of the Movie table with a different table?

1. SELECT M.Title, A.Name

FROM Movie M, Actor A  
WHERE M.ActorID = A.ActorID;

1. SELECT M.Title, A.Name

FROM Movie M, Actor A  
WHERE M.ActorID = A.MovieID;

1. SELECT M.Title, A.Name

FROM Movie M

RIGHT JOIN Actor A  
ON M.ActorID = A.ActorID;

1. SELECT M.Title, A.Actor FROM Movie M  
   INNER JOIN Actor A ON M.ActorID = A.ActorID

This question is asking which statement illustrates an OUTER JOIN, not an INNER JOIN. The only 2 statements above that include outer joins are b and c (LEFT JOIN and RIGHT JOIN).

B is an invalid statement, so C has to be the answer. Please review this document to get a good understanding of outer versus inner joins:

[Left joins / Right Joins / Inner Joins - Examples](https://westerngovernorsuniversity-my.sharepoint.com/:w:/g/personal/maria_schenk_wgu_edu/Eafty6p0RX9PhR5ytuM4igUB1ykNDPOm3eurRHGlA0U7HQ?e=R5Ujvj)

23.

Assume there are two tables, A and B.  
Which rows will always be included in the result set if Table A is inner joined with Table B?

a. Only rows in Tables A and B that share the join condition

b. All rows in Table B

c. All rows in Table A

d. Only rows in Tables A and B that do not share the join condition.

24.

The database contains a table named Book.  
Write a SQL query to return all data from the Book table without directly referencing any column names.

**SELECT \* FROM Book;**

25.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL query to retrieve the Title and Genre values for all records in the Book table with a Year value of 2020. Ensure your result set returns the columns in the order indicated.

**SELECT Title, Genre**

**FROM Book**

**WHERE Year = 2020;**

26.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL query to display all Title values in alphabetical order A–Z.

**SELECT Title**

**FROM Book**

**ORDER BY Title ASC;**

**SELECT Title**

**FROM Book**

**ORDER BY Title;**

This question is testing to see if the student knows how to use the ORDER BY clause. Here is the general syntax:

A group of black text

Description automatically generated

And here is the correct answer.

*Continued below*

27.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL query to output the unique Genre values and the number of books with each genre value from the Book table as GenreCount. Sort the results by the Genre in alphabetical order A–Z. Ensure your result set returns the columns in the order indicated.

In this case you are being asked to count the number of Movies having each type of Genre. For this reason, you must use GROUP BY to group all moves with each Genre together and then do **a count. The attribute you are grouping on must be listed first.**

**SELECT Genre, COUNT(\*) AS GenreCount  
FROM Book  
GROUP BY Genre  
ORDER BY Genre ASC;**

Please take a look at these resources:

[GROUP BY](https://www.w3schools.com/sql/sql_groupby.asp)   
[COUNT](https://www.w3schools.com/sql/sql_count_avg_sum.asp)

28.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

The YearSales table has the following columns:  
Year—integer  
TotalSales—bigint unsigned  
Releases—integer  
Write a SQL query to display both the Title and the TotalSales (if available) for all books. Ensure your result set returns the columns in the order indicated.

Here, you must have a JOIN because you are pulling a value from each of the tables.

**SELECT Title, TotalSales   
FROM Book LEFT JOIN YearSales  
ON Book.Year = YearSales.Year;**

29.

The Book table has the following columns:  
ID—integer, primary key, auto\_increment  
Title—variable-length string  
Genre—variable-length string  
Year—integer

Write a SQL query to return how many books have a Year value of 2019.

**SELECT COUNT(\*)**

**FROM Book**

**WHERE Year = 2019;**

30. A(n) \_\_\_\_\_ is a query that is embedded (or nested) inside another query.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a. | alias | b. | operator |
|  | c. | subquery | d. | view |

31. All changes to a table structure are made using the \_\_\_\_\_ command, followed by a keyword that produces the specific changes a user wants to make.

1. ALTER TABLE
2. UPDATE TABLE
3. COMMIT TABLE

32. The SQL aggregate function that gives the number of rows containing non-null values for a given column is \_\_\_\_\_.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a. | COUNT | b. | MIN |
|  | c. | MAX | d. | SUM |

|  |  |  |
| --- | --- | --- |
| 33. The \_\_ON\_\_\_ condition of a JOIN is generally composed of an equality comparison between the foreign key and the primary key of related tables.  34. Which kind of relationship is displayed in the entity-relationship diagram below?   1. Binary one-to-one 2. Unary many-to-many 3. Ternary one-to-one 4. Binary one-to-many 5. Unary one-to-one  |  |  | | --- | --- | |  |  | |
| A picture containing text, screenshot, font, number  Description automatically generated |

35. Which kind of relationship is displayed in the entity-relationship diagram below?

1. Binary one-to-one
2. Unary many-to-many
3. Ternary one-to-one
4. Binary many-to-many
5. Unary one-to-one

A picture containing text, screenshot, font, number

Description automatically generated

36. Which kind of relationship is displayed in the entity-relationship diagram below?

1. Binary one-to-one
2. Unary many-to-many
3. Ternary one-to-one
4. Binary many-to-many
5. Unary one-to-one

A diagram of a work flow

Description automatically generated with low confidence

37. Which data type can be designated to allow for storage of dates

1. VARCHAR
2. INTEGER
3. DECIMAL
4. TIMESTAMP/DATE

38.

A picture containing text, font, number, screenshot

Description automatically generated

Which query would produce a result set that listed all of the patients, regardless of whether they had an appointment in the Exam table or not.

1. SELECT patient\_id, first\_name, last\_name, birthdate, exam\_date

FROM Patient RIGHT JOIN EXAM

ON Patient.patient\_id = Exam.patient\_id;

1. SELECT patient\_id, first\_name, last\_name, birthdate, exam\_date

FROM Patient INNER JOIN EXAM

ON Patient.patient\_id = Exam.patient\_id;

1. SELECT patient\_id, first\_name, last\_name, birthdate, exam\_date

FROM Patient INNERJOIN EXAM

ON Patient.patient\_id = Exam.exam\_id;

1. SELECT patient\_id, first\_name, last\_name, birthdate, exam\_date

FROM Patient LEFT JOIN EXAM

ON Patient.patient\_id = Exam.patient\_id;

39.

A picture containing text, screenshot, font, businesscard

Description automatically generated

Which query would produce a result set that listed all of the Model Types, regardless of whether a model was assigned to that type.

1. SELECT model\_id, lname, Models.model\_type\_id, hourly\_fee

FROM Models

RIGHT JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;

1. SELECT model\_id, lname, Models.model\_type\_id, hourly\_fee

FROM Models

LEFT JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;

1. SELECT model\_id, lname, Models.model\_type\_id, hourly\_fee

FROM Models

INNER JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;

1. SELECT model\_id, lname, Models.model\_type\_id, hourly\_fee

FROM Models

NATURAL JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;

40.

Write a query to list Models along with the model type they are assigned. Include all model information and the hourly fee.

**SELECT model\_id, lname, birthdate, Models.model\_type\_id, hourly\_fee**

**FROM Models**

**INNER JOIN Model\_Type  
ON Models.model\_type\_id = Model\_Type.model\_type\_id;**

41.

Write a query to delete Amy Lin from the Customer table shown below.

A picture containing text, number, screenshot, font

Description automatically generated

**DELETE FROM Customer**

**WHERE CustomerID = 101;**

42. Write a SQL statement to update Blanca Garcia’s phone number to 555-222-1234.

**UPDATE Customer**

**SET Mobile Phone = ‘555-222-1234’**

**WHERE ID = 104;**

A picture containing text, number, screenshot, font

Description automatically generated

43. Write a SQL statement to retrieve all of the Customers from Seattle from the table below.

A picture containing text, number, screenshot, font

Description automatically generated

**SELECT \***

**FROM Customer**

**WHERE City = ‘Seattle’;**

44. Write the CREATE TABLE statement to create the Exam table below. Make sure to designate the primary key and foreign key of the table.

A picture containing text, screenshot, font, number

Description automatically generated

**CREATE TABLE Exam (**

**exam\_id INT,**

**exam\_date DATE,**

**exam\_reason VARCHAR(255),**

**patient\_id INT,**

**PRIMARY KEY (exam\_id),**

**FOREIGN KEY (patient\_id) REFERENCES Patient(patient\_id)**

**);**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bagel ID** | **Bagel Name** | **Bagel Description** | **Cost/Bagel** |
| RB | Radically Raisin Bagel | Raisin Bagel | 2.00 |
| EB | All the Stuff Bagel | Everything Bagel | 2.35 |
| CB | Chip off the Block Bagel | Chocolate Chip Bagel | 2.25 |
| MB | Keepin’ it Healthy Bagel | Multigrain Bagel | 2.00 |
| NB | Classic New York Bagel | New York Bagel | 1.75 |

45. Write a query to pull all of the data from the Bagel table below. Order the result set alphabetically by Bagel ID.

**SELECT \***

**FROM Bagel**

**ORDER BY Bagel ID ASC;**

46. Write a query to count the number of customers in each city.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer ID** | **First Name** | **Last Name** | **Address** | **City** | **State** | **Zip** |
| 101 | Amy | Lin | 123 A Street | Seattle | WA | 98118 |
| 102 | Viya | Prasad | 111 B Avenue | Seattle | WA | 98118 |
| 104 | Blanca | Garcia | 8674 Main Street | Portland | OR | 97212 |
| 107 | Shelia | Brown | 22815 Foster Road | Portland | OR | 97217 |
| 108 | Anna | Cruz | 444 Apple Lane | Troutdale | OR | 97708 |
| 110 | Amy | Smith | 39878 Acorn Place | Chicago | IL | 60435 |

**SELECT City, COUNT(\*)**

**FROM Customer**

**GROUP BY City;**

47. Write a query that will return all of the cities from the Customer table above. Do not include duplicates -list each city only once.

**SELECT DISTINCT City**

**FROM Customer;**

48. Write a query to list customers from Chicago or Seattle.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer ID** | **First Name** | **Last Name** | **Address** | **City** | **State** | **Zip** |
| 101 | Amy | Lin | 123 A Street | Seattle | WA | 98118 |
| 102 | Viya | Prasad | 111 B Avenue | Seattle | WA | 98118 |
| 104 | Blanca | Garcia | 8674 Main Street | Portland | OR | 97212 |
| 107 | Shelia | Brown | 22815 Foster Road | Portland | OR | 97217 |
| 108 | Anna | Cruz | 444 Apple Lane | Troutdale | OR | 97708 |
| 110 | Amy | Smith | 39878 Acorn Place | Chicago | IL | 60435 |

**SELECT \***

**FROM Customer**

**WHERE City = ‘Chicago’ OR City = ‘Seattle’;**

49. Write a join statement that pulls the values listed from 3 of the tables below (Volunteer, Registration, and Activity Registration).

VolunteerID

LastName

Registration Date

ActivityID

A picture containing text, screenshot, font, number

Description automatically generated

**SELECT Volunteer.VolunteerID, LastName, Registration.Registration\_Date, Activity.Activity\_ID,**

**FROM Volunteer JOIN Registration**

**ON Volunteer.volunteerID=Registration.volunteerID**

**JOIN Activity\_Registration**

**ON RegistrationRegistrationID=Activity\_Registration.RegistrationID;**

50. Write a join statement that pulls the values listed from 3 of the tables below (Customer, Invoice, and Invoice Item).

CustomerID

InvoiceID

Date

ProductID

A diagram of a company

Description automatically generated

**SELECT Customer.CustomerID, Invoice.InvoiceID, Date, Invoice\_Item.ProductID**

**FROM Customer JOIN Invoice**

**ON Customer.CustomerID=Invoice.CustomerID**

**JOIN Invoice\_Item**

**ON  Invoice.ProductID=Invoice\_Item.ProductID;**

51. Write a join statement that pulls the values listed from 3 of the tables below (Models, Model Types, Phone Numbers).

model\_id

last\_name  
first\_name

model\_type\_id

phone\_number

A picture containing text, screenshot, font, number

Description automatically generated

**SELECT Model.model\_id, last\_name, first\_name, Model\_Types.model\_type\_id, phone\_number**

**FROM Models JOIN ModelTypes**

**ON Models.model\_type\_id = Model\_Types.model\_type\_id**

**JOIN PhoneNumbers**

**ON  Models.model\_id = Phone\_Numbers.model\_id;**

52.

**student\_grades**

|  |  |  |
| --- | --- | --- |
| studentID | lastName | grade |
| 44444 | Jones | B |
| 24352 | Chang | A |
| 98724 | Wu | B |
| 98789 | Chatham | A |
| 98887 | Fredricks | C |
| 00878 | Moore | A |
| 22078 | Prasad | C |
| 79692 | Perez | A |

Refer to the given SELECT statement and the table above.  
  
SELECT grade, COUNT(\*) AS gradeCount   
FROM student\_grades  
GROUP BY grade;

Which clause added to the statement finds grades with a count greater than 2?

a. WITH gradeCount <> DUPLICATE

b. NOT gradeCount = NULL

c. **HAVING gradeCount > 2**

d. INCLUDING gradeCount > 2

53.

**SELECT grade, COUNT(\*)  
FROM student\_grades  
GROUP BY grade;**

54. Write a query to find the highest model fee.

SELECT MAX(hourly\_fee) FROM Models;



55.

**UPDATE Model\_Types  
SET hourly\_fee = hourly\_fee + 100  
WHERE hourly\_fee < 600;**

56. C

57. B

58.

ALTER TABLE dog CHANGE [COLUMN] name name VARCHAR(25);

<https://dev.mysql.com/doc/refman/8.0/en/alter-table.html>