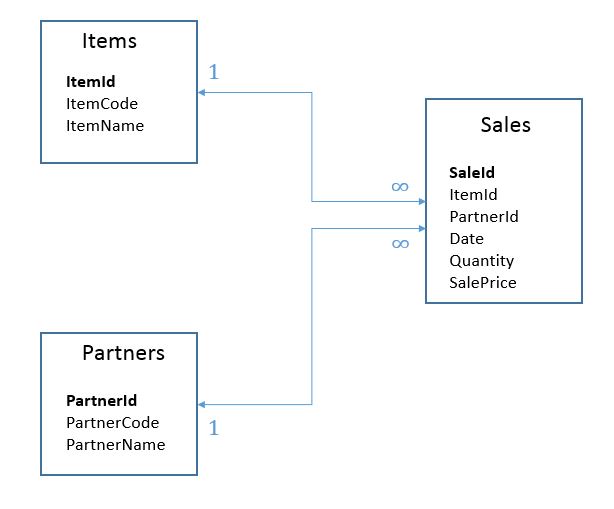
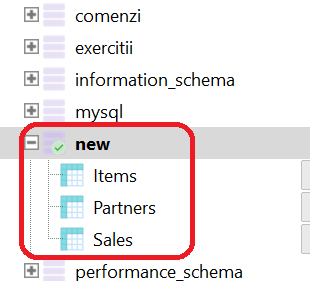
We have the following diagram:

The first column from each table represents the primary key.

Sample

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Items** | | |  |  |  |  |  |  |  |
| ItemId | ItemCode | ItemName |  |  |  |  |  |  |  |
| 1 | DF56R | Laptop HP |  |  |  |  |  |  |  |
| 2 | SER6G | Imprimanta |  |  |  |  |  |  |  |
| 3 | FG78R | Mouse |  |  |  |  |  |  |  |
| 4 | MNJ4L | Laptop Dell |  | **Sales** | | | | | |
| 5 | SS4F56 | Laptop Asus |  | **SaleId** | **ItemId** | **PartnerId** | **Date** | **Quantity** | **SalePrice** |
|  |  |  |  | 890734 | 1 | 100 | 2010-01-03 | 2 | 234 |
|  |  |  |  | 890734 | 3 | 100 | 2010-01-03 | 2 | 12 |
|  |  |  |  | 890735 | 4 | 234 | 2011-03-05 | 1 | 100 |
|  |  |  |  | 890735 | 1 | 234 | 2011-03-05 | 1 | 234 |
| **Partners** | | |  | 890736 | 3 | 789 | 2012-11-07 | 3 | 12 |
| **PartnerId** | **PartnerCode** | **PartnerName** |  |  |  |  |  |  |  |
| 100 | PC4567 | Gigel |  |  |  |  |  |  |  |
| 101 | AS456 | Vasilica |  |  |  |  |  |  |  |
| 102 | PC679 | Mitica |  |  |  |  |  |  |  |
| 103 | NY584 | Bobita |  |  |  |  |  |  |  |
| 104 | NY222 | Florica |  |  |  |  |  |  |  |

1. Write a query that will show the **total number** of ItemId.
2. Write a query that will show the **total number** of SaleId for each PartnerId.
3. Write a query that will show all the PartnerId for which the PartnerCode begin with NY
4. Write a query that will show the total SalePrice paid by PartnerId=100
5. Write a query that will show the Average SalePrice on March 2010.
6. Write a query that will return the **Total Sales Value** for each ItemId from Items table for JANUARY 2014.
7. Write a query that will return the **Total Sales Value** registered to the parner: “Gigel” for January 2010.
8. Write a query that will return the top 10 best-selling products from a value standpoint.
9. Delete all partners from the Partners table that don’t have any sales associated.
10. Modify the sales price for item having the ItemId **3** for all sales in the last 5 days for the partner: “Gigel”. The new sales price is **2550.**
11. Insert a new partner record in the Partners table having the PartnerId **105** and the PartnerName“**Veronica**”.
12. If we apply an index on the **Sales** table based on the column “Date”, do we increase, decrease or is there no effect on the performance of the query from point 1 of this test? Please explain why.
13. If we apply an index on the **Partners** table based on the column “PartnerCode” do we increase, decrease or is there no effect on the performance of the insert query from point 11 of this test? Please explain why.
14. Add the column MU (Measure Unit) in the table **Items**.
15. Delete the previously created column from the table **Items.**
16. Please list at least 2 differences between the functions **COALESCE** and **ISNULL**.
17. Please list at least 3 differences between **TRUNCATE**  and  **DELETE.**
18. What is the difference between **UNION** and **UNION ALL.**
19. How many clustered indexes can be defined on one table?
20. Which id value will be returned by the below select statement given the following details:

DECLARE @Logic TABLE (ID INT, Product VARCHAR(50))

INSERT INTO @Logic

VALUES (1, 'Baseball Hat'),

(2, 'Bicycle'),

(3, 'Snowboard'),

(4, 'Goggles'),

(5, 'Shows')

SELECT ID

FROM @Logic

WHERE Product = 'Bicycle' OR Product = 'Snowboard' AND ID = 4

Answer:

1. Having the following table structure what do the operators **INTERSECT** and **EXCEPT** return?

|  |  |  |
| --- | --- | --- |
| **Table\_A** |  | **Table\_B** |
|  |  |  |
| **Col\_X** |  | **Col\_Y** |
| 18 |  | 1 |
| 20 |  | 20 |
| 4 |  | 2 |
| 1 |  | 3 |
| 2 |  |  |

SELECT Col\_X AS **'Intersect'**

FROM Table\_A

INTERSECT

SELECT Col\_Y

FROM Table\_B

Answer:

SELECT Col\_X AS **'Except'**

FROM Table\_A

EXCEPT

SELECT Col\_Y

FROM Table\_B

Answer:

1. Please specify the value returned by each of the statements below:

DECLARE @a VARCHAR(10), @b VARCHAR(10), @c VARCHAR(10), @d VARCHAR(20), @e VARCHAR(20)

SET @a= 'ana'

SET @b= ' are mere'

SET @c= ' are mere '

SELECT LEN(@a) AS V1 Answer:

SELECT LEN(@b) AS V2 Answer:

SELECT LEN(@c) AS V3 Answer:

SELECT @d = @a + @c Answer:

SELECT LEN(@d) AS V4 Answer:

SELECT @e = LTRIM(RTRIM(@a)) + LTRIM(RTRIM(@c)) Answer:

SELECT LEN(@e) AS V5 Answer:

Answer:

*-- 1. Write a query that will show the total number of ItemId.*

**SELECT** **count**(ItemId) total\_number\_items

**FROM** Items a;

*-- 2. Write a query that will show the total number of SaleId for each PartnerId.*

**SELECT** **count**(a.SaleId) total\_number\_sales, a.PartnerId

**FROM** Sales a

**GROUP** **BY** a.PartnerId;

*-- 3. Write a query that will show all the PartnerId for which the PartnerCode*

*-- begin with NY*

**SELECT** a.PartnerId

**FROM** Partners a

**WHERE** a.PartnerCode **LIKE** 'NY%';

*-- 4. Write a query that will show the total SalePrice paid by PartnerId=100*

**SELECT** **SUM**(a.SalePrice) total\_SalePrice, a.PartnerId

**FROM** Sales a

**WHERE** a.PartnerId = 100;

*-- 5. Write a query that will show the Average SalePrice on March 2010.*

**SELECT** **AVG**(a.SalePrice)

**FROM** Sales a

**WHERE** **DATE** **between** '2010-03-01' **AND** '2010-03-31';

*-- 6. Write a query that will return the Total Sales Value for each ItemId from Items*

*-- table for JANUARY 2014.*

*-- Am inlocuit 2014 cu 2010 pentru a vedea ceva date*

*# Varianta in care verificam pentru absolut toate item-urile*

**SELECT** a.ItemId, **SUM**(b.SalePrice) total\_sales\_value

**FROM** Items a

**LEFT** **JOIN** Sales b

**ON** a.ItemId = b.ItemId **AND** b.**DATE** **between** '2010-01-01' **AND** '2010-01-31'

**GROUP** **BY** a.ItemId;

*# Varianta in care verificam doar pentru item-urile cu vanzari*

**SELECT** a.ItemId, **SUM**(b.SalePrice) total\_sales\_value

**FROM** Items a

**inner** **JOIN** Sales b

**ON** a.ItemId = b.ItemId

**WHERE** b.**DATE** **between** '2010-01-01' **AND** '2010-01-31'

**GROUP** **BY** a.ItemId;

*-- 7. Write a query that will return the Total Sales Value registered to the*

*-- parner: “Gigel” for January 2010.*

**SELECT** b.PartnerName, **SUM**(a.SalePrice) total\_sales\_value

**FROM** Sales a

**INNER** **JOIN** Partners b **ON** a.PartnerId = b.PartnerId

**WHERE** b.PartnerName **LIKE** 'Gigel'

**AND** a.**DATE** **BETWEEN** '2010-01-01' **AND** '2010-01-31'

**GROUP** **BY** b.PartnerName;

*-- 8. Write a query that will return the top 10 best-selling products from a VALUE*

*-- standpoint.*

*# fiind doar 3 item-uri, le afiseaza pe toate 3*

**SELECT** a.ItemId, **SUM**(a.SalePrice) total\_value

**FROM** Sales a

**GROUP** **BY** a.ItemId

**ORDER** **BY** total\_value **DESC**

**LIMIT** 10;

*-- 9. Delete all partners from the Partners table that don’t have any sales associated.*

**DELETE** a

**SELECT** \* **FROM** Partners a

**left** **JOIN** Sales b **ON** a.PartnerId = b.PartnerId

**WHERE** b.PartnerId **IS** **NULL**;

*#check*

**SELECT** \*

**FROM** Partners a

**left** **JOIN** Sales b

**ON** a.PartnerId = b.PartnerId

**WHERE** b.PartnerId **IS** **not** **NULL**;

*-- 10. Modify the sales price for item having the ItemId 35 for all sales in the*

*-- last 5 days for the partner: “Gigel”. The new sales price is 2550.*

*# am schimbat ItemId 35 cu 3, caci nu avem 35 in tabel*

**UPDATE** Sales a

**INNER** **JOIN** Partners b

**ON** a.PartnerId = b.PartnerId

**SET** a.SalePrice = 2550

**WHERE** a.ItemId = 3 **AND** b.PartnerName = 'Gigel'

**AND** (a.**DATE** **between** **NOW**() - **INTERVAL** 5 **day** **AND** **NOW**() - **INTERVAL** 1 **DAY**);

*-- 11. Insert a new partner record in the Partners table having the PartnerId 105*

*-- and the PartnerName “Veronica”.*

**INSERT** **INTO** Partners

**VALUES** (105, 'VE445', 'Veronica');

*#check*

**SELECT** \* **FROM** Partners;

*-- 12. If we apply an index on the Sales table based on the column “Date”, DO*

*-- we increase, decrease or is there no effect on the performance of the query FROM*

*-- point 1 of this test? Please explain why.*

*/\**

*un index nou pe coloana Date nu va schimba cu nimic viteza de executie al exercitiului 1*

*intrucat nu avem nicio conditie de filtrare pe coloana Date*

*\*/*

*-- 13. If we apply an index on the Partners table based on the column “PartnerCode”*

*-- do we increase, decrease or is there no effect on the performance of the INSERT*

*-- query from point 11 of this test? Please explain why.*

*#initial era mentionat exercitiul 6, insa la 11 se face insertul*

*/\**

*Se va ingreuna executia insertului, intrucat va trebui sa scrie noua informatie in index*

*\*/*

*-- 14. Add the column MU (Measure Unit) in the table Items.*

**ALTER** **TABLE** Items

**ADD** MU **VARCHAR**(5);

*-- 15. Delete the previously created column from the table Items.*

**ALTER** **TABLE** Items

**DROP** **COLUMN** MU;

*-- 16. Please list at least 2 differences between the functions COALESCE and ISNULL.*

*# COALESCE(column, replacement) - replacement has the same type as the column type*

*# ISNULL(column, replacement) - replacement can be of any type*

*# COALESCE can be used in any database system (it is ANSI standard) while ISNULL*

*#works for some of them.*

*# COALESCE accepts many inputs and will display the first non null value, but*

*#ISNULL accepts only 2 inputs*

*-- 17. Please list at least 3 differences between TRUNCATE and DELETE.*

*#Delete can remove one row or more based on a condition while truncate deletes all at once*

*/\**

*TRUNCATE este mai rapid decat DELETE*

*E mai rapid pentru ca nu scrie in Loguri nimic*

*DELETE se poate face prin filtrarea randurilor (where)*

*\*/*

*-- 18. What is the difference between UNION and UNION ALL.*

*#The difference between UNION and UNION ALL is that UNION does not have duplicates*

*#in the result while UNION ALL accepts duplicates.*

*-- 19. How many clustered indexes can be defined on one TABLE?*

*#Only one clustered index can be define in one Table because the table data is stored*

*#in the same order as the data in the index, so it can be only one order.*

*-- 20. Which id value will be returned by the below select statement given the*

*-- following details:*

**DECLARE** @Logic **TABLE** (ID **INT**, Product **VARCHAR**(50))

**INSERT** **INTO** @Logic

**VALUES** (1, 'Baseball Hat'),

(2, 'Bicycle'),

(3, 'Snowboard'),

(4, 'Goggles'),

(5, 'Shows')

**SELECT** ID

**FROM** @Logic

**WHERE** Product = 'Bicycle' **OR** Product = 'Snowboard' **AND** ID = 4

*#Answer: ID = 2*

*-- 21. Having the following table structure what do the operators INTERSECT*

*-- and EXCEPT RETURN?*

**SELECT** Col\_X **AS** 'Intersect'

**FROM** Table\_A

INTERSECT

**SELECT** Col\_Y

**FROM** Table\_B

*#Answer: Intersect returns only the common VALUES from Col\_X AND Col\_Y*

**SELECT** Col\_X **AS** 'Except'

**FROM** Table\_A

**EXCEPT**

**SELECT** Col\_Y

**FROM** Table\_B

*#Answer: EXCEPT returns the values from col\_X that are different from col\_y*

*-- 22. Please specify the value returned by each of the statements below:*

**DECLARE** @a **VARCHAR**(10), @b **VARCHAR**(10), @c **VARCHAR**(10), @d **VARCHAR**(20), @e **VARCHAR**(20)

**SET** @a= 'ana'

**SET** @b= ' are mere'

**SET** @c= ' are mere '

**SELECT** LEN(@a) **AS** V1 Answer:3

**SELECT** LEN(@b) **AS** V2 Answer:9

**SELECT** LEN(@c) **AS** V3 Answer:9

**SELECT** @d = @a + @c Answer:'ana are mere '

**SELECT** LEN(@d) **AS** V4 Answer:12

**SELECT** @e = **LTRIM**(**RTRIM**(@a)) + **LTRIM**(**RTRIM**(@c)) Answer:'anaare mere'

**SELECT** LEN(@e) **AS** V5 Answer:11