Assignment 4

Due date: March 22nd, 2023

Total: 10 points

Querying using Joins

Create the database

Execute the sql script that is attached with this assignment and ensure that all tables are created and seeded with the data.

Queries

Open a new query window and target the sales table. Save this query window as "Assignmet4.sql". All the scripts you will be executing will run from this file. To project the data of the commonalities between two tables, the relational concept of joins in MySQL is expressed through the join keyword. Your first task is to implement several of these types of joins to produce the following results.

1. Project the number of items sold and total revenue for Sales Rep and Each item. This will be a simple join between the ledger, salesrep and the item tables matching on iditem and Idrep.

SalesRep	ItemName	Total	Revenue
Bruce Wayne	Batarang	327	3270
Bruce Wayne	Batmobile	2	39000002
Bruce Wayne	Ray Gun	5	7500
Bruce Wayne	Batcycle	1	2500000
Bruce Wayne	Super Suit	22	7700
Bruce Wayne	Justice League Pin	1805	90250
Bruce Wayne	Batwallet	3711	742200
Bruce Wayne	Flash Cleaning Se...	279	279000
Bruce Wayne	Adamantium Rage	32	160448

2. Project the Total Number of items sold for each sales rep by item, with the highest revenue as the beginning record and the lowest as the last record. Your output should look like the following, naturally with more records than this:

Salesrep	ItemName	Total	Revenue
Bruce Wayne	Batmobile	2	39000002
Logan Howlett	Batmobile	2	39000002
Diana Prince	Batmobile	1	19500001
Hal Jordan	Batmobile	1	19500001
Barry Allen	Batmobile	1	19500001
Barbara Gordan	Batmobile	1	19500001
Logan Howlett	Batwallet	45	5250400
Hal Jordan	Batcycle	2	5000000
Barry Allen	Batcycle	2	5000000

3. Project the Total Sales for each region with the Highest sales and quantity sold as the first record and lowest at the bottom. Your output should look like the following, with the more records than shown:

Region	ItemName	Quantity	TotalSales
Gotham	Batmobile	7	136500007
Metropolis	Batmobile	4	78000004
Star City	Batmobile	3	58500003
Themascura	Batmobile	1	19500001
Gotham	Batcycle	7	17500000
Gotham	Batwallet	73198	14639600
Metropolis	Batcycle	5	12500000
Metropolis	Batwallet	53355	10671000
Star City	Batcycle	3	7500000

4. Get the Total number of sales and items sold by each rep for each month ordering by Year, month, and from Greatest Revenue to lowest Revenue:

SYear	SMonth	Rep	Total	Revenue
2019	January	Hal Jordan	10	3016690
2019	January	Logan Howlett	8	749822
2019	January	Barbara Gordan	12	721404
2019	January	Diana Prince	9	706940
2019	January	Bruce Wayne	3	273000
2019	January	Barry Allen	4	230150
2019	February	Hal Jordan	13	19888051
2019	February	Barbara Gordan	13	799400
2019	February	Barry Allen	7	395510

Using Views

Create a view called vwRepSales with the following criteria:

Syear - A calculated Year of the purchase date

SMonth – A calculated Month of the purchase date

Purchase_date – the purchase date from the ledger

Idrep – the rep id

Rep – The concatenated first and last name of the sales Rep

ItemName – The item name of the purchased item

Total - The sum of the items sold from the ledger for each item

Revenue – The sum of the number of items multiplied by the item cost

Make sure the view is properly grouped so that the results accurately reflect the purchases based on purchase date.

A select * from vwRepSales should look like the following.

SYear	SMonth	purchase_date	idrep	Rep	ItemName	Total	Revenue
2019	11	2019-11-30 00:00:00	1	Bruce Wayne	Batarang	327	3270
2019	9	2019-09-22 00:00:00	1	Bruce Wayne	Batmobile	1	19500001
2019	7	2019-07-25 00:00:00	1	Bruce Wayne	Batmobile	1	19500001
2019	4	2019-04-08 00:00:00	1	Bruce Wayne	Ray Gun	2	3000
2019	11	2019-11-14 00:00:00	1	Bruce Wayne	Ray Gun	1	1500
2019	8	2019-08-26 00:00:00	1	Bruce Wayne	Ray Gun	2	3000
2019	8	2019-08-18 00:00:00	1	Bruce Wayne	Batcycle	1	2500000
2019	7	2019-07-02 00:00:00	1	Bruce Wayne	Super Suit	5	1750
2019	7	2019-07-20 00:00:00	1	Bruce Wayne	Super Suit	2	700
2019	9	2019-09-12 00:00:00	1	Bruce Wayne	Super Suit	3	1050
2019	8	2019-08-06 00:00:00	1	Bruce Wayne	Super Suit	2	700
2019	4	2019-04-26 00:00:00	1	Bruce Wayne	Super Suit	5	1750
2019	4	2019-04-14 00:00:00	1	Bruce Wayne	Super Suit	5	1750

From this view solve the following:

1. Get All the sales by month between January 1, 2019 and May 30th, 2019 ordered by the month name. Project only the month name and the total Revenue for each month.

MonthName	Revenue
April	26942483
May	5821784
March	2395022
January	5698006
February	21683561

2. Get All the sales by month and user between January 1, 2019 and May 30th, 2019 order by Month, then Total Revenue Descending. Project only the Month name, Rep, and total revenue.

MonthName	Rep	Revenue
January	Hal Jordan	3016690
January	Logan Howlett	749822
January	Barbara Gordan	721404
January	Diana Prince	706940
January	Bruce Wayne	273000
January	Barry Allen	230150
February	Hal Jordan	19888051
February	Barbara Gordan	799400
February	Barry Allen	395510

3. Get All the sales by month and user between May 1st, 2019 and December 31st, 2019 order by Sales Rep ascending then Month descending.

Rep	MonthName	Revenue
Barbara Gordan	December	689748
Barbara Gordan	November	396330
Barbara Gordan	October	20348621
Barbara Gordan	September	507250
Barbara Gordan	August	1035670
Barbara Gordan	July	405500
Barbara Gordan	June	407700
Barbara Gordan	May	2919890
Barry Allen	December	325850

Using Functions

Create a Function called salesbonus with the following criteria:

1. The function will take in a revenue of type Decimal(10,2).
2. The function will calculate the bonus by multiplying .15 * the revenue
3. The function will return the bonus as a decimal(10,2)

Use the function to get all sales by month with the bonus. Your output should look like the following:

MonthName	Rep	Revenue	bonus
April	Bruce Wayne	107300	16095.00
May	Bruce Wayne	334950	50242.50
March	Bruce Wayne	369700	55455.00
January	Bruce Wayne	273000	40950.00
February	Bruce Wayne	5000	750.00
May	Barry Allen	456014	68402.10
February	Barry Allen	395510	59326.50
April	Barry Allen	2837750	425662.50
March	Barry Allen	289900	43485.00
January	Barry Allen	230150	34522.50
February	Logan Howlett	314450	47167.50
March	Logan Howlett	271690	40753.50
April	Logan Howlett	19841981	2976297.15

Using Stored Procedures

Create a stored procedure with the following criteria:

1. Have the following input parameters
    a. Repid int
    b. Startdate varchar(20)
    c. EndDate varchar(20)
2. No output parameters
3. The stored procedure shall use the vwREpSales to get the month, Sales Rep and Revenue and bonus based on the sales rep id provided and the date range entered.
    a. If no salesrep is not provided, the stored procedure shall list all sales reps.
4. Calling the stored procedure using the following statements will produce the table preceding it.

    CALL `sales`.`RepSales`(null, '2019-1-1', '2019-3-31');

MonthName	Rep	Revenue	bonus
March	Bruce Wayne	369700	55455.00
January	Bruce Wayne	273000	40950.00
February	Bruce Wayne	5000	750.00
February	Barry Allen	395510	59326.50
March	Barry Allen	289900	43485.00
January	Barry Allen	230150	34522.50
February	Logan Howlett	314450	47167.50
March	Logan Howlett	271690	40753.50
January	Logan Howlett	749822	112473.30
January	Hal Jordan	3016690	452503.50
February	Hal Jordan	19888051	2983207.65
March	Hal Jordan	271350	40702.50
February	Diana Prince	281150	42172.50
March	Diana Prince	815850	122377.50

CALL `sales`.`RepSales`(1, '2019-1-1', '2019-3-31');

MonthName	Rep	Revenue	bonus
March	Bruce Wayne	369700	55455.00
January	Bruce Wayne	273000	40950.00
February	Bruce Wayne	5000	750.00

ERD Design

Create an design for the following Scenario:

Although you always wanted to be an artist, you ended up being an expert on databases because you love to cook data and you somehow confused *database* with *data paste*. Your old love is still there, however, so you set up a database company, `ArtBase`, that builds a product for art galleries. The core of this product is a database with a schema that captures all the information that galleries need to maintain. Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art. For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, lithograph, sculpture, photograph), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits, still lifes, works by Picasso, or works of the 19th century; a given piece may belong to more than one group. Each group is identified by a name (like those just given) that describes the group. Finally, galleries keep information about customers. For each customer, galleries keep that person's unique name, address, total amount of dollars spent in the gallery, and the artists and groups of art that the customer tends to like.

1. Identify the relational schema for the database. Make sure to clearly indicate primary keys and foreign keys.
   Ex: **ARTIST** (a_name, a_birthplace, a_age, a_style_of_art)
2. Create an ERD for the scenario.

Submission

Submit Assignment4.sql with all implementation scripts and an Assignment4.docx file with the ERD and relational Schema on D2L under Assignment 4.

Assessment

2 Points – Queries

2 Points – View Creation

2 Points-Function creation

2 Points – Stored procedure creation

2 Points -ERD design

Learning Outcomes

This assignment is designed to demonstration the use of views to summarize data to simplify sql queries. It also introduces the student to the use of functions and stored procedures to do some dynamic database programming. Finally, it demonstrates the creation of an ERD from a scenario as well as the relational schema.

Disclaimer

Please review the syllabus on academic integrity and the submission policy. I will follow both strictly, so please adhere to the policy for each subsequent assignment or project.