**Project 1**

**TASK 1**

**A. Make a list of the entities for the above process**

The following are the entities for the process:

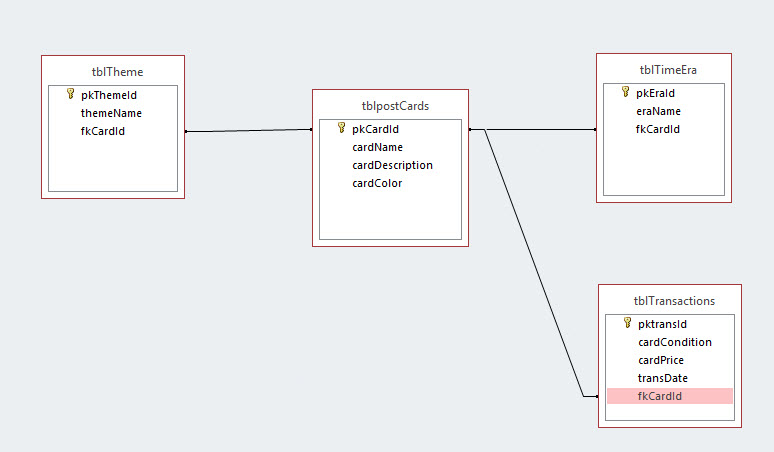
1. tblpostCards
2. tblTimeEra
3. tblTheme
4. tblTransactions

**B. For each entity define the required attributes and the data type of each of those attributes.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Attribute** | **Data Type** | **Key/Relationship** |
| tblpostCards | pkCardid | int | Primary Key |
| cardName | varchar(50) |  |
| cardDescription | text |  |
| cardColor | varchar(10) |  |
| tblTimeEra | pkEraId | int | Primary Key |
| eraName | varchar(20) |  |
| fkCardId | int | Foreign Key refers to primary key of tblpostcCards |
| tblTransactions | pktransId | int | Primary Key |
| cardCondition | varchar(10) |  |
| cardPrice | float |  |
| transDate | date |  |
| fkCardId | int | Foreign Key refers to primary key of tblpostcCards |
| tblTheme | pkThemeId | Int | Primary Key |
| themeName | Varchar(20) |  |
| fkCardId | int | Foreign Key refers to primary key of tblpostcCards |

**TASK 2**

**A. Design the database schema (diagram) using arrows to define the references, based on the description. Your schema can be handwritten at this point.**



**B. Explain why you took specific decisions regarding:**

* **The relations you have created.**

A post card will have a unique ID number which can be used as reference for that post card hence, I created the relation based on the ID. Post Card name and color maybe same as mentioned in the definition and only ID remains unique.

* **The cardinality of the relationships.**

1. The cardinality of relationship between tblpostCards and tblTransactions will be One-One as for a particular post card transaction is done only once.
2. The cardinality of relationship between tblpostCards and tblTimeEra is One-One as one post card belongs to only one time era.
3. The cardinality of relationship between tblpostCards and tblTheme is One-Many as a post card can be listed in many themes.

**C. Based on your decisions, make a list of the constraints that are posed on your schema, mentioning the type of each of the constraints.**

* The constraints will be on both the relations.
* Both will have OnUpdate and OnDelete constraint.

**TASK 3**

1. **Write down SQL code to define the above mentioned database schema.**

* CREATE TABLE tblpostCards

( pkCardId int NOT NULL AUTO\_INCREMENT, cardName varchar(50), cardDescription text, cardColor varchar(10)

PRIMARY KEY (pkCardId)

);

* CREATE TABLE tblTimeEra

( pkEraId int NOT NULL AUTO\_INCREMENT, eraName varchar(20), fkCardId int

PRIMARY KEY (pkEraId)

FOREIGN KEY (fkCardId) REFRENCES tblpostCards(pkCardId)

);

* CREATE TABLE tblTransactions

( pktransID int NOT NULL AUTO\_INCREMENT, cardCondition varchar(15), cardPrice float, transDate date, fkCardId int

PRIMARY KEY (pktransId)

FOREIGN KEY (fkCardId) REFRENCES tblpostCards(pkCardId)

);

* CREATE TABLE tblTheme

( pkThemId int NOT NULL AUTO\_INCREMENT, themeName varchar(20), fkCardId int,

PRIMARY KEY (pkThemeId)

FOREIGN KEY (fkCardId) REFRENCES tblpostCards(pkCardId)

);

1. **SQL Statements to add data.**

* INSERT INTO `postcards`.`tblpostcards` (`pkCardid`, `cardName`, `cardDescription`, `cardColor`) VALUES (NULL, 'Romance in Paris - 1939i', 'Vintage Card from Paris.', 'White');
* INSERT INTO `postcards`.`tbltransactions` (`pktransId`, `cardCondition`, `cardPrice`, `transDate`, `fkCardId`) VALUES (NULL, 'acceptable', '3', '2014-09-16', '2');
* INSERT INTO `postcards`.`tbltheme` (`pkThemeId`, `themeName`, `fkCardId`) VALUES (NULL, 'Buildings', '3');
* INSERT INTO `postcards`.`tbltimeera` (`pkEraId`, `eraName`, `fkCardId`) VALUES (NULL, 'Golden Era', '1');

**TASK 4**

**Based on the above database go on and create appropriate queries for the following:**

**a. Show all the postcards that George has in his collection**

**SQL Code:**

[select](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html) pktransId, cardName, cardCondition from tblpostcards, tbltransactions where tblpostcards.pkCardid=tbltransactions.fkCardId;

**Output:**

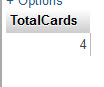
****

**b. Show the number of postcards George has in his collection**

**SQL Code:**

[select](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html) [count](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fgroup-by-functions.html#function_count)(\*) as TotalCards from tbltransactions;

**Output:**

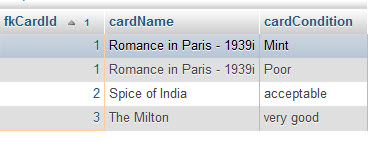


**c. Show the unique postcards George has in his collection**

**SQL Code:**

[select](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html) distinct(tbltransactions.fkCardId), tblpostcards.cardName, tbltransactions.cardCondition from tbltransactions, tblpostcards where tbltransactions.fkCardId=tblpostcards.pkCardid order by tbltransactions.fkCardId;

**Output:**

****

**d. Show the duplicate postcards George has in his collection**

**SQL Code:**

SELECT COUNT('tbltransactions.fkCardId') AS 'count', 'cardName' FROM tblpostcards, tbltransactions WHERE tbltransactions.fkCardId=tblpostcards.pkCardid HAVING 'count' > 1 ORDER BY 'count'

**Output:**

**e. Show the postcards purchased after a given year (you decide) with a thematic category “people”**

**SQL Code:**

[select](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html) tbltransactions.fkCardId, tblpostcards.cardName, tbltheme.themeName, tbltransactions.transDate from tbltransactions, tbltheme, tblpostcards where tbltransactions.fkCardId=tbltheme.fkCardId [and](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html#operator_and) tbltransactions.fkCardId=tblpostcards.pkCardId [and](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html#operator_and) transDate >'2014-08-01\*' [and](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html#operator_and) tbltheme.themeName='People';

**Output:**

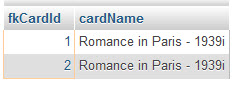


**f. Show the postcards purchased belonging into the golden era which are of thematic category “people” or “buildings”.**

**SQL Code:**

select distinct(tbltransactions.fkCardId), tblpostcards.cardName from tbltransactions, tbltheme, tbltimeera, tblpostcards where tbltransactions.fkCardId=tbltheme.fkCardId and tbltransactions.fkCardId=tbltimeera.fkCardId and tbltimeera.eraName='Golden Era' and tbltheme.themeName='People' or tbltheme.themeName='Building' and tblpostcards.pkCardid=tbltransactions.fkCardId group by tbltransactions.fkCardId;

**Output:**



**g. Show a list with George’s multi-themed postcards**

**SQL Code:**

[select](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html) tblpostcards.pkCardid, tblpostcards.cardName, tbltheme.themeName from tblpostcards, tbltheme where tbltheme.fkCardId=tblpostcards.pkCardid;

**Output:**

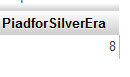


**h. Show how much money George has paid so far for silver era postcards.**

**SQL Code:**

[select](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html) [sum](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fgroup-by-functions.html#function_sum)(tbltransactions.cardPrice) as PiadforSilverEra from tbltransactions, tbltimeera where tbltransactions.fkCardId=tbltimeera.fkCardId [and](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html#operator_and) tbltimeera.eraName='Silver Era';

**Output:**



**i. Show the average price of the postcards for each different condition category.**

**SQL Code:**

select tbltransactions.cardCondition, avg(tbltransactions.cardPrice) as avgPrice from tbltransactions group by tbltransactions.cardCondition;

**Output:**



**j. Show the largest amount of money paid for one postcard by George, which is not in “poor” condition**

**SQL Code:**

select max(cardPrice) as MaximumPaid from tbltransactions where tbltransactions.cardCondition != 'Poor';

**Output:**

