University of Makati

College of Computing and Information Sciences

ELEC 5 – Group Activity

GROUP-4-INTELLITECH Members:
Bantilo, Jade Daniele M.
Corda, Ryan P.
Derige, Paul Angelo
Eugenio, Shiloh B.

Scenario: INTELLITECH Vince's Vinyl (A Music Record Shop)

Vince is eager to get going. Just today he had a customer come in and sell him a dozen old albums. One is quite rare and could be worth a lot of money. Vince doesn't want to lose track of it. He is ready to get organized and start entering his transactions in the database. You review your design with him and promise that you will begin building the database immediately. But, you remind him, it is important to test the database before actually starting to use it for the business.

1. Review your diagram for the database making sure that the design is complete and normalized.

2. Create the database in SQL Server

3. Create the tables in the new database, selecting appropriate data types for the columns, setting a primary key for each table, and setting allow nulls as appropriate

4. Create a database diagram and create the relationships among tables

5. Add some sample data to each table

6. **Documentation:** Make a Data Dictionary that lists each table, all the columns for that table, the data types for each column.

Rubrics

For Nos. 1, 4, & 6 (45 points)

Problem Solving	Excellent 5 pts	Good 4 pts	Average 3 pts	Fair 2 pts
Content	Excellent Appropriate content is used for each problem. Student clearly understands the mathematical concepts.	Good Appropriate content is used for each problem. Student shows some understanding of the mathematical concepts.	Average Appropriate content may be used. Student shows little understanding of the mathematical concepts.	Fair Appropriate content is not observed. Student does not demonstrate an understanding of the mathematical concepts.
Solution/Organization	Excellent The solution is written in clear and coherent way. Solution is presented in a very organized manner.	Good The solution is written in clear and coherent way.	Average The solution is not written in clear and coherent way.	Fair The solution is not written in clear and coherent way, or may not be observed.
Accuracy	Excellent Solution is very clear and accurate.	Good Solution is clear and accurate.	Average Solution is somehow clear and correct.	Fair Solution is not clear and may not be correct.

For Nos. 2, 3, & 5 (15 points)

Criteria	Mastery (2.5 points)	Meeting (1.5 points)	Does Not Meet (0.5 point)
Scientific Accuracy	Use of multiple variables that demonstrate understanding of environmental factors.	Use of some variables that demonstrate understanding of environmental factors.	Use of variables that fail to demonstrate understanding of environmental factors.
Coding Efficiency	Code is easy to follow and direct.	Code is mostly easy to follow and direct.	Code is convoluted and unnecessarily long.

https://forum.code.org/t/share-your-rubric-for-assessing-computer-models/2927/48

1. Review your diagram for the database making sure that the design is complete and normalized.

2. Create the database in SQL Server

```
CREATE DATABASE vince_vinyl;
```

3. Create the tables in the new database, selecting appropriate data types for the columns, setting a primary key for each table, and setting allow nulls as appropriate

albums table

```
CREATE TABLE albums (
    album_id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    artist VARCHAR(255) NOT NULL,
    genre VARCHAR(100),
    release_year INT,
    `condition` VARCHAR(50),
    price DECIMAL(10, 2),
    is_rare BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

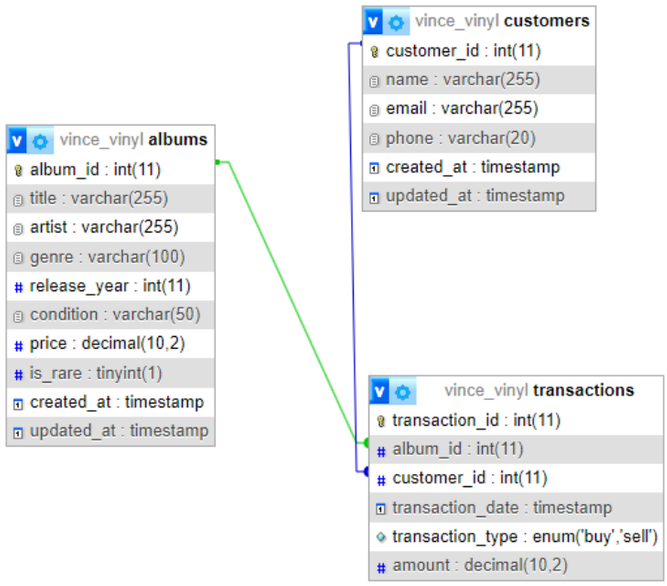
customers table

```
CREATE TABLE customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255),
    phone VARCHAR(20),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

transactions table

```
CREATE TABLE transactions (
    transaction_id INT AUTO_INCREMENT PRIMARY KEY,
    album_id INT,
    customer_id INT,
    transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    transaction_type ENUM('buy', 'sell') NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (album_id) REFERENCES albums(album_id),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

4. Create a database diagram and create the relationships among tables



5. Add some sample data to each table

Insert data in table albums

```sql
INSERT INTO albums (title, artist, genre, release_year, `condition`, price, is_rare) VALUES
('Liwanag sa Dilim', 'Rivermaya', 'Pop Rock', 1998, 'Good', 1500.00, TRUE),
('Puso', 'Yeng Constantino', 'Pop', 2011, 'Excellent', 2000, TRUE),
('Ako... Dapat', 'Gloc-9', 'Hip-Hop', 2013, 'Very Good', 900.00, FALSE),
('Balisong', 'Mojofly', 'Alternative Rock', 2001, 'Good', 1000.00, FALSE),
('Himig natin', 'Pink Floyd', 'Rock', 1973, 'Excellent', 2000.00, TRUE);
```

```sql
SELECT * FROM `albums`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	album_id	title	artist	genre	release_year	condition	price	is_rare	created_at	updated_at
Edit Copy Delete	1	Liwanag sa Dilim	Rivermaya	Pop Rock	1998	Good	1500.00	1	2024-09-06 14:13:25	2024-09-06 14:13:31
Edit Copy Delete	2	Puso	Yeng Constantino	Pop	2011	Excellent	2000.00	1	2024-09-06 14:13:25	2024-09-06 14:13:34
Edit Copy Delete	3	Ako... Dapat	Gloc-9	Hip-Hop	2013	Very Good	900.00	0	2024-09-06 14:13:25	2024-09-06 14:13:36
Edit Copy Delete	4	Balisong	Mojofly	Alternative Rock	2001	Good	1000.00	0	2024-09-06 14:13:25	2024-09-06 14:13:39
Edit Copy Delete	5	Himig natin	Pink Floyd	Rock	1973	Excellent	2000.00	1	2024-09-06 14:13:25	2024-09-06 14:13:41

Insert data in table customers

```sql
INSERT INTO customers (name, email, phone) VALUES
('Shiloh Eugenio', 'shiloheugenio@gmail.com', '09155483547'),
('Jade Daniele Bantilo', 'jadebantilo@gmail.com', '09642547591'),
('Ryan Corda', 'ryancorda@gmail.com', '09412548723'),
('Paul Angelo Derige', 'paulangeloderige@gmail.com', '09171234567'),
('Arriane Camille Pamintuan', 'arrianecamille@gmail.com', '09182345678')

```

```sql
SELECT * FROM `customers`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	customer_id	name	email	phone	created_at	updated_at
Edit Copy Delete	1	Shiloh Eugenio	shiloheugenio@gmail.com	09155483547	2024-09-06 14:20:22	2024-09-06 14:20:22
Edit Copy Delete	2	Jade Daniele Bantilo	jadebantilo@gmail.com	09642547591	2024-09-06 14:20:22	2024-09-06 14:20:22
Edit Copy Delete	3	Ryan Corda	ryancorda@gmail.com	09412548723	2024-09-06 14:20:22	2024-09-06 14:20:22
Edit Copy Delete	4	Paul Angelo Derige	paulangeloderige@gmail.com	09171234567	2024-09-06 14:20:22	2024-09-06 14:20:22
Edit Copy Delete	5	Arriane Camille Pamintuan	arrianecamille@gmail.com	09182345678	2024-09-06 14:20:22	2024-09-06 14:20:22

Insert data in transactions table

```
INSERT INTO transactions (transaction_id, album_id, customer_id, transaction_date, transaction_type, amount) VALUES
(1, 1, 1, '2024-09-01', 'buy', 1500.00),
(2, 2, 2, '2024-09-02', 'buy', 2000.00),
(3, 3, 3, '2024-09-03', 'sell', 900.00),
(4, 4, 1, '2024-09-04', 'buy', 1000.00),
(5, 5, 2, '2024-09-05', 'sell', 2000.00);
```

```
SELECT * FROM `transactions`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	transaction_id	album_id	customer_id	transaction_date	transaction_type	amount
Edit Copy Delete	1	1	1	2024-09-01 00:00:00	buy	1500.00
Edit Copy Delete	2	2	2	2024-09-02 00:00:00	buy	2000.00
Edit Copy Delete	3	3	3	2024-09-03 00:00:00	sell	900.00
Edit Copy Delete	4	4	1	2024-09-04 00:00:00	buy	1000.00
Edit Copy Delete	5	5	2	2024-09-05 00:00:00	sell	2000.00

6. **Documentation:** Make a Data Dictionary that lists each table, all the columns for that table, the data types for each column.

Albums		
Column	**Type**	**Description**
album_id	int(11)	The unique identifier for the album.
title	varchar(255)	The title of the album.
artist	varchar(255)	The name of the artist who created the album.
genre	varchar(100)	The genre or category of music the album falls under.
release_year	Int(11)	The year the album was released.
condition	varchar(50)	The physical condition of the album (e.g., New, Used, Mint).
price	decimal(102)	The price of the album.
is_rare	tinyint(1)	A flag indicating whether the album is considered rare (1 for yes, 0 for no).
created_at	timestamp	The timestamp when the album record was created.
updated_at	timestamp	The timestamp when the album record was last updated.

Customers		
Column	**Type**	**Description**
customer_id	int(11)	The unique identifier for the customer.
name	varchar(255)	The name of the customer.
email	varchar(255)	The email address of the customer.
phone	varchar(20)	The phone number of the customer.
created_at	timestamp	The timestamp when the customer record was created.
updated_at	timestamp	The timestamp when the customer record was last updated.

Transaction		
Column	**Type**	**Description**
transaction_id	int(11)	The unique identifier for the transaction.
album_id	int(11)	The identifier for the album involved in the transaction,

		referencing the album_id from the Albums table.
customer_id	int(11)	The identifier for the customer involved in the transaction, referencing the customer_id from the Customers table.
transaction_date	timestamp	The date and time when the transaction occurred.
transaction_type	enum(‘buy’,’sell’)	The type of transaction, either 'buy' (purchase) or 'sell' (sale).
amount	decimal(102)	The amount of money involved in the transaction.