COMP3278A Introduction to Database Management Systems 2023-24

Assignment 1

Due Date: Oct 12 23:59

Question 1 (50%). Please read the below background information of a simple blockchain data structure.

Background Information. A simple blockchain is a linked list of blocks as shown in the below figure, where each block contains a unique block ID, a hash (a unique string of characters), a hash of the previous block, and a creation time of the block. Each block is used to store transaction data, which are records of transferring digital currency. A unit of the digital currency is called a token. Each token can be uniquely identified by a token ID and has a market value.
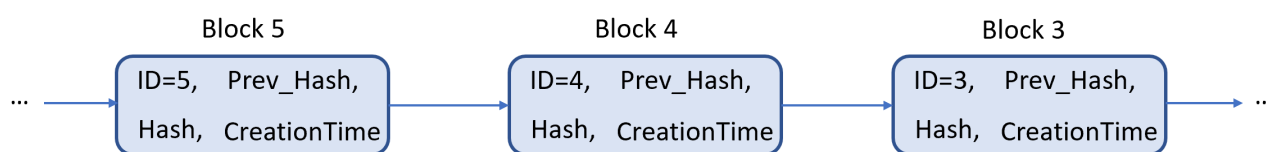
Please read the below users' requirements, and answer questions (a) and (b).

User Requirements. A HKU-DBMS team is building a transaction system by using blockchain.
- The main purpose of the system is to manage transactions of tokens between accounts.
- Each user in the system is identified by a unique username. Each user has a profile attribute, which consists of a name, an email address, and the date the user joined the system.
- A user owns one or more accounts. Each account in the system is identified by a unique account number, and an account must be owned by a user.
- There are two specifications of accounts, including token account and virtual account. An account is either a token account or a virtual account.
- Token account is an account that keeps tokens in the system. Each token can be uniquely identified by a token ID and has its market value.
- A token mush be owned by only one token account. A token account may or may not have one or more tokens.
- Virtual account has no token by itself. Instead, it must be linked to one or more token accounts. A token account may or may not have multiple virtual accounts.
- A transaction is initiated by specifying a total amount to be transferred from one account to another account, for example, either from a token account to a virtual account or from a virtual account to a token account. Each transaction is uniquely identified by a transaction ID. It is possible to leave (attach) several messages along with the transaction.
- A transaction must be kept in only one block. Each block has a unique block ID and a creation time, and each block may or may not have multiple transactions. A hash value is calculated and recorded in each block.
- A block is always linked to only one previous block (except for the very first block in the system), in order to form a blockchain. The hash value of the previous block is also explicitly recorded in a block. It is possible that multiple blocks have the same previous block at the same time.
- In fact, each transaction is completed by using one or more internal transactions. An internal transaction has an internal ID, which can only uniquely identify each internal transaction when combining with the corresponding transaction ID.
- Each internal transaction is used to transfer one token from one token account to another token account.

(a) Please model the above users' requirements by drawing the corresponding E-R Diagram.
- If a requirement is unclear, a reasonable assumption could be made. You must **clearly state** all such assumptions.
(b) Please reduce your E-R Diagram into relational table schemas. For each relation, underline the primary key and identify all foreign key(s), if any.

Question 2 (50%). Please read the below story, ER diagram, and table schema, and then answer the query tasks by writing SQL.

APC Printing Company (APC) is a company specialized in producing prints on different media. The company wants to setup an online printing ordering system to allow customer to order their prints online. Their in-house developers have designed the ER diagram and produced a relational table schema with data type assigned. You are hired to join them in preparing the database for the system.

The E-R diagram is shown and explained as follows.
- A user of the printing ordering system (AppUser) can be a customer or an admin user.
- Albums are created by the users in the system.
- Image can be uploaded to an album, each identified by an image_id and the corresponding album_id.
- Customers can place orders in the system (PrintOrder). The total amount of an order is calculated from the items in the order.
- An item is an image from an album.
- A product should be chosen for each item be printed on.
- The quantity of each item in an order could be specified by the customer, a subtotal can be calculated from the product price and specified quantity.
- An admin will be assigned to each order as the person-in-charge (PIC). A history of order status will be maintained by PIC.
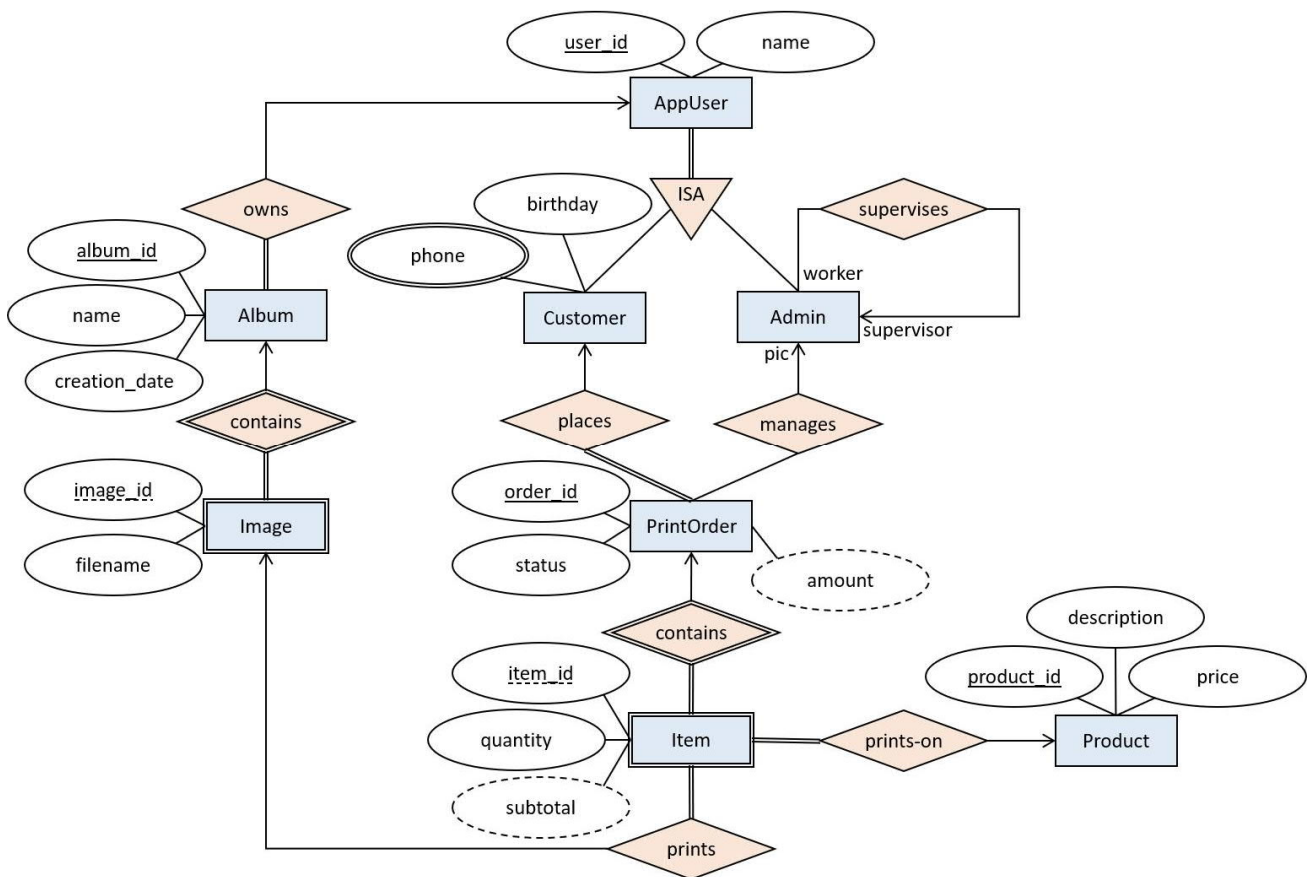- Some admin users may be assigned some other admin users as supervisors.

Table schema (with data type)

AppUser (user_id: INT, name: VARCHAR(80))
Foreign keys: *none*

Customer (user_id: INT, birthday: DATE)
Foreign keys: user_id refers **AppUser**

CustomerPhone (user_id: INT, phone: VARCHAR(20))
Foreign keys: user_id refers **Customer**

Admin (user_id: INT, supervisor_id: INT)
Foreign keys: user_id refers **AppUser**, {supervisor_id} refers **Admin** {user_id}

Album (album_id: INT, name: VARCHAR(80), creation_date: DATE, user_id: INT)
Foreign keys: user_id refers **AppUser**

Image (album_id: INT, image_id: INT, filename: VARCHAR(80))
Foreign keys: album_id refers **Album**

Product (product_id: INT, description: TEXT, price: decimal(10,2))
Foreign keys: *none*

PrintOrder (order_id: INT, user_id: INT, status: VARCHAR(80), pic_id: INT)
Foreign keys: user_id refers **Customer**, {pic_id} refers **Admin** {user_id}

Item (order_id: INT, item_id: INT, quantity: INT, product_id: INT,
album_id: INT, image_id: INT)
Foreign keys: order_id refers **PrintOrder**, product_id refers **Product**,
{album_id, image_id} refers **Image**

For each of the following tasks from 1) to 10), give a single SQL query that provide the result needed.
-   Select at least the primary key(s) of all related tables if the columns to be listed is not specified.
-   Assume ascending order if ordering direction is not specified.
-   Your query must handle all possible scenario according to the given constraint.
-   There should not be duplicated results.
-   **Q2_test_tables.sql** and **Q2_test_data.sql** on Moodle are sample data for your reference (more details about the sample data are at the end of this file).

1) List all available products in the system ordered by price.
2) Find the user IDs and names of all customers.
3) Find the customer name, phone numbers, and order ID of all orders that has "New" in the status. If the customer has multiple phone numbers, concatenate them by commas in the result. (**Hint**: use GROUP_CONCAT)
4) List all items of order ID 3, showing the item ID, quantity, product description, and the subtotal. Subtotal equals to the product price times quantity.
5) List all orders by customer with user ID 5, showing the order ID, status, number of items (sum of quantity), and the total amount.
6) Find the names of all users that is a customer and an admin user.
7) Find all admin users is not a supervisor.
8) List all albums owned by admin users, in a descending order of the number of images in the album. If there exist two albums with the same number of images, order them in descending order of the

creation date.
9) Find the number of times each of the images in album ID 1 is included in an order. An image that is placed in the same order multiple times should be counted as once only. Images that has never been ordered must also be shown.
10) Find all customer name of orders that has the greatest number of order items (not considering quantity).

Submission Requirement. Please submit your work in a single PDF file on Moodle before the deadline. Late penalty will be applied for late submission. For each of the tasks in question 2, the **SQL query** and the **query results** of MySQL (e.g. screenshot) should be provided.

Supplementary. Sample data in **Q2_test_tables.sql** and **Q2_test_data.sql**.

AppUser

user_id	name
1	Gebhard Nasato
2	Sunitha Aslan
3	Kemuel Sala
4	Harley Shine
5	Marcos Lannon
6	Simonides Lauritsen
7	Fearghas Hamm
8	Rati Boon
9	Dylan Bellomi
10	Saoirse Meyer
11	Sievert Leclair
12	Emiel Crouch
13	Inzhu McCoy
14	Adsila Fields
15	Lucija Romero

CustomerPhone

user_id	phone
1	937-0692
2	698-7147
5	712-1926
7	906-4850
7	953-1400
8	429-1754
10	529-5867
11	481-3146
14	824-2009
15	246-6855
15	497-5406

Customer

user_id	birthday
1	1985-07-14
2	1988-03-21
5	1990-01-04
7	1992-10-11
8	1990-05-04
10	2004-12-01
11	1979-01-24
14	1987-08-03
15	1994-11-01

Image

album_id	image_id	filename
1	1	IMG_1204.jpg
1	2	IMG_1206.jpg
1	3	IMG_1207.jpg
1	4	IMG_1211.jpg
1	5	IMG_1215.jpg
2	1	DSC_0011.jpg
2	2	DSC_0012.jpg
2	3	DSC_0018.jpg
2	4	DSC_0020.jpg
3	1	IMG_0099.jpg
3	2	IMG_0130.jpg
3	3	IMG_0151.jpg
4	1	IMG_20191231_00014.jpg
4	2	IMG_20191231_00018.jpg
4	3	IMG_20200101_00002.jpg
5	1	IMG_0167.jpg
5	2	IMG_0174.jpg
5	3	IMG_0188.jpg
6	1	DSC_1024.jpg
6	2	DSC_1031.jpg
6	3	DSC_1033.jpg
6	4	DSC_1040.jpg
6	5	DSC_1044.jpg

Item

order_id	item_id	quantity	product_id	album_id	image_id
1	1	1	1	2	1
1	2	1	1	2	3
1	3	1	1	2	4
1	4	4	3	2	4
2	1	3	1	4	1
2	2	2	1	4	2
3	1	5	1	1	2
3	2	1	1	1	4
3	3	2	6	1	3
4	1	5	5	1	2
4	2	1	4	1	1
5	1	1	3	5	2
5	2	3	2	5	1
5	3	3	4	5	3
5	4	1	5	5	3
6	1	1	3	1	2
6	2	3	2	1	2
6	3	3	4	5	3
6	4	1	5	5	3
7	3	1	1	4	1
7	99	1	2	4	3

PrintOrder

order_id	user_id	status	pic_id
1	2	Waiting for collection.	1
2	10	Printing in progress.	4
3	5	Payment pending	1
4	15	New order	NULL
5	5	New order	NULL
6	5	New order	NULL
7	10	New order	NULL

Admin

user_id	supervisor_id
1	NULL
6	NULL
13	NULL
2	1
3	1
4	2
9	6
14	6

Abum

album_id	name	creation_date	user_id
1	Stock photos	2019-11-21	1
2	Family reunion	2019-11-25	3
3	Christmas 2019	2019-12-30	5
4	Party time	2020-01-04	10
5	New year trip 2020	2020-01-20	5
6	Scenic photos	2020-02-01	13

Product

product_id	description	price
1	4R photo	1.20
2	5R photo	2.30
3	8R photo	5.00
4	12R photo	15.00
5	2R photo card	2.50
6	Photo mug	230.00

[End of Paper]