

CS3402 Database Systems

Assignment 2

Question A SQL. [60 points]

Given the following database about the information of a Health and Fitness Application. It structures the management of user profiles, health records, exercise sessions, and sharing of those exercise activities.

- **User** (**user_id**: integer, **name**: string, **email**: string, **age**: integer, **gender**: string)

// Describe the basic information about users, including their ID, name, email, age, and gender.

Number	Column Name	Column Type	Not Null	Default Value	PK
0	user_id	INT	1		1
1	name	VARCHAR(50)	1		0
2	email	VARCHAR(50)	1		0
3	age	INT	1		0
4	gender	VARCHAR(1)	1		0

- **HealthRecord** (**record_id**: integer, **user_id**: integer, **date**: date, **weight**: float, **height**: float, **heart_rate**: integer)

// Records detailed health metrics for users, such as weight, height, and heart rate, linking these records to the corresponding user by user_id.

Number	Column Name	Column Type	Not Null	Default Value	PK
0	record_id	INT	1		1
1	user_id	INT	1		0
2	date	DATE	1		0
3	weight	FLOAT	1		0
4	height	FLOAT	1		0
5	heart_rate	INT	1		0

- **ExerciseType** (**type_id**: integer, **name**: string)

// Describes different types of exercises available in the application, each with a unique ID and name.

Number	Column Name	Column Type	Not Null	Default Value	PK
0	type_id	INT	1		1
1	name	VARCHAR(50)	1		0

- **ExerciseSession** (**session_id**: integer, **user_id**: integer, **type_id**: integer, **date**: date, **duration**: integer, **calories_burned**: integer)

// Describes individual exercise sessions, including the type of exercise, duration, calories burned, and the session date. It links to both the user and the specific type of exercise.

Number	Column Name	Column Type	Not Null	Default Value	PK
0	session_id	INT	1		1
1	user_id	INT	1		0
2	type_id	INT	1		0
3	date	DATE	1		0
4	duration	INT	1		0
5	calories_burned	INT	1		0

- **ActivityShare** (**share_id**: integer, **user_id**: integer, **session_id**: integer, **share_date**: date)

// Describes instances where users share their exercise sessions, including the date of sharing. It references both the user who shared the session and the specific exercise session shared.

Number	Column Name	Column Type	Not Null	Default Value	PK
0	share_id	INT	1		1
1	user_id	INT	1		0
2	session_id	INT	1		0
3	share_date	DATE	1		0

(Note: You need to log into <http://dboj.cs.cityu.edu.hk:8080/> and choose SQL Homework 2 under Exercise to answer the questions for the online judge. Submissions directly made to Canvas will not be graded.)

-- 1. Find the exercise session IDs for the user 'Bob Jones'. [6 points]

```
SELECT session_id FROM ExerciseSession
WHERE user_id = (SELECT user_id FROM User WHERE name = 'Bob Jones');
```

-- 2. List the names of users who have not logged any exercise sessions. [6 points]

```
SELECT name FROM User
WHERE user_id NOT IN
(
    SELECT DISTINCT user_id FROM ExerciseSession
);
```

-- 3. List the names of users who have logged more than one exercise session on any given day.

```

SELECT U.name
FROM User U
WHERE U.user_id IN (
    SELECT ES.user_id
    FROM ExerciseSession ES
    GROUP BY ES.user_id, ES.date
    HAVING COUNT(*) > 1
);

```

-- 4. Calculate the average calories burned of male users. [6 points]

```

SELECT AVG(calories_burned) AS avg_calories_burned
FROM ExerciseSession
WHERE user_id IN (
    SELECT user_id FROM User WHERE gender = 'M'
);

```

-- 5. List name(s) of the exercise type(s) that have never been shared.

```

SELECT ET.name
FROM ExerciseType ET
WHERE ET.type_id NOT IN (
    SELECT DISTINCT E.type_id
    FROM ExerciseSession E, ActivityShare AS ASH
    WHERE E.session_id = ASH.session_id
);

```

-- 6. List the ID of each exercise type along with its frequency of occurrence (i.e., the total number of times the exercise session has been done), in descending order by frequency. If a type has never been performed in any exercise session, it will not be included in the result. [6 points]

```

SELECT type_id, COUNT(*) AS frequency
FROM ExerciseSession
GROUP BY type_id
ORDER BY frequency DESC;

```

-- 7. List the names of users who have logged at least two exercise sessions but have never shared any of exercise sessions. [6 points]

```

SELECT U.name
FROM User U
WHERE U.user_id IN (
    SELECT ES.user_id

```

```

FROM ExerciseSession ES
GROUP BY ES.user_id
HAVING COUNT(ES.session_id) >= 2
) AND U.user_id NOT IN (
    SELECT ASH.user_id FROM ActivityShare AS ASH
);

```

-- 8. List the user ID(s) with the largest weight change in the HealthRecord. [6 points]

```

SELECT HR.user_id
FROM HealthRecord AS HR
GROUP BY HR.user_id
HAVING MAX(HR.weight) - MIN(HR.weight) = (
    SELECT MAX(max_weight - min_weight) FROM (
        SELECT
            MAX(weight) AS max_weight,
            MIN(weight) AS min_weight
        FROM HealthRecord
        GROUP BY user_id
    ) AS WeightDifference
);

```

-- 9. List the user name(s) with the maximum number of health records. [6 points]

```

SELECT name
FROM User
WHERE user_id IN (
    SELECT user_id
    FROM HealthRecord
    GROUP BY user_id
    HAVING COUNT(*) = (
        SELECT MAX(record_count)
        FROM (
            SELECT user_id, COUNT(*) as record_count
            FROM HealthRecord
            GROUP BY user_id
        ) AS RecordCounts
    )
);

```

-- 10. List the name(s) of exercise type(s) that result in the highest average calorie burn for female users only. [6 points]

```

SELECT ET.name
FROM ExerciseType ET
WHERE ET.type_id IN (
    SELECT ES.type_id
    FROM ExerciseSession ES
    WHERE ES.user_id IN (
        SELECT user_id FROM User WHERE gender = 'F'
    )
)
GROUP BY ES.type_id
HAVING AVG(ES.calories_burned) = (
    SELECT MAX(avg_calories_burned)
    FROM (
        SELECT AVG(ES2.calories_burned) as avg_calories_burned
        FROM ExerciseSession ES2
        WHERE ES2.user_id IN (SELECT user_id FROM User WHERE gender = 'F')
        GROUP BY ES2.type_id
    ) AS max_avg_calories_burned
);

```

Question B Relational Algebra [40 points]

Considering the relational database schema of an album retailers shown below, write the following queries using relational **algebra expressions** (Note: No score will be given for SQL).

ALBUM

<u>Album_ID</u>	Album_Name	Artist_Name	Release_Year	Genre
-----------------	------------	-------------	--------------	-------

ARTIST

<u>Artist_Name</u>	Gender	Birthplace	Birthyear
--------------------	--------	------------	-----------

SONG

<u>Album_ID</u>	<u>Song_Name</u>	Artist_ID
-----------------	------------------	-----------

SELLING_RECORD

<u>Album_ID</u>	Inventory	Sell_Amount	Price
-----------------	-----------	-------------	-------

CUSTOMER_RECORD

<u>Album_ID</u>	<u>Customer_ID</u>	Date	Price	Discount
-----------------	--------------------	------	-------	----------

- Find the names of the artists who have album with the genre of "Pop music". [6 points]

$$\pi_{\text{Artist_Name}} (\sigma_{\text{Genre} = \text{"Pop music"}} (\text{ALBUM} * \text{ARTIST}))$$

$$\text{OR } \pi_{\text{Artist_Name}} (\sigma_{\text{Genre} = \text{"Pop music"}} \text{ALBUM})$$

- List the name of albums released in 2023 by the Hong Kong artist. [6 points]

$$\text{ARTIST_HK} = \sigma_{\text{BirthPlace} = \text{"Hong Kong"}} (\text{ARTIST})$$

$$\text{ALBUM_2023} = \sigma_{\text{ReleaseYear} = 2023} (\text{ALBUM})$$

$$\pi_{\text{Album_Name}} (\text{ARTIST_HK} * \text{ALBUM_2023})$$

- 3) List the name of female artist and the total selling amount of all its album. [6 points]

$$\begin{aligned} \text{ALBUM_FEMALE} &= \sigma_{\text{Gender}=\text{"female"}}(\text{ALBUM} * \text{ARTIST}) \\ \text{RESULT} &\leftarrow \pi_{\text{Artist_Name}} \bowtie_{\text{SUM Sell_Amount}} (\text{SELLING_RECORD} * \text{ALBUM_FEMALE}) \end{aligned}$$

- 4) Retrieve the Album ID which contains the maximum number of songs. [6 points]

$$\begin{aligned} \text{SONG_NO} &\leftarrow \pi_{\text{Album_ID}} \bowtie_{\text{COUNT Song_Name}} (\text{SONG}) \\ \text{RESULT} &\leftarrow \pi_{\text{Album_ID}} (\pi_{\text{Album_ID}} \bowtie_{\text{MAX COUNT(Song_Name)}} (\text{SONG_NO})) \end{aligned}$$

- 5) Find the Customer ID who buy more than three albums with the genre of "Folk music" in last year. (Note: from "2023-01-01" to "2023-12-31") [8 points]

$$\begin{aligned} \text{LAST_YEAR_RECORDS} &\leftarrow \sigma_{\text{Date} \geq \text{"2023-01-01"} \text{ and Date} \leq \text{"2023-12-31"}} (\text{CUSTOMER_RECORD}) \\ \text{FOLK_ALBUM} &\leftarrow \sigma_{\text{Genre}=\text{"Folk music"}} (\text{ALBUM}) \\ \text{FILTERED_RECORDS} &\leftarrow \text{LAST_YEAR_RECORDS} * \text{FOLK_ALBUM} \\ \text{CNO} &\leftarrow \sigma_{\text{COUNT(AlbumID)} > 3} (\pi_{\text{Customer_ID}} \bowtie_{\text{COUNT AlbumID}} (\text{FILTERED_RECORDS})) \\ &\quad \pi_{\text{Customer_ID}} \text{CNO} \end{aligned}$$

- 6) Find the Customer ID who buy all the album of the artist "Eason Chan". [8 points]

$$\begin{aligned} \text{EASON_ALBUM} &\leftarrow \pi_{\text{Album_ID}} (\sigma_{\text{Artist_Name}=\text{"Eason Chan"}} (\text{ALBUM})) \\ \text{EASON_FAN} &\leftarrow \pi_{\text{Album_ID}, \text{Customer_ID}} (\text{CUSTOMER_RECORD}) \\ \text{RESULT} &\leftarrow \text{EASON_FAN} \div \text{EASON_ALBUM} \end{aligned}$$