1. Database schema (database = nusbank)
   1. **Users** Table
      1. **User\_id : PRIMARY KEY to be unique**

|  |  |  |
| --- | --- | --- |
| Field | datatype | Description |
| user\_id | INT UNSIGNED | User Unique number (PRIMARY KEY) |
| name | VARCHAR(50) | User name |
| mail | VARCHAR(50) | User email |
| mobile | VARCHAR(50) | User mobile |
| nric | VARCHAR(50) | User national IC |

* 1. **Accounts** Table
     1. **Acct\_numer as PRIMARY KEY to be unique**

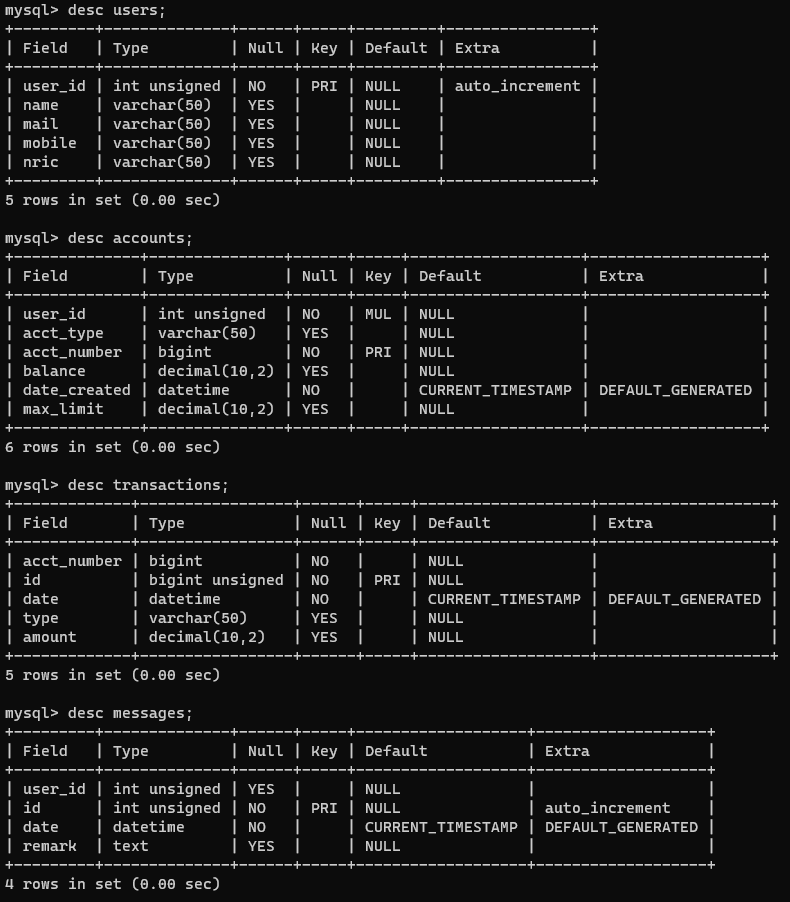
|  |  |  |
| --- | --- | --- |
| Field | Datatype | Description |
| user\_id | INT UNSIGNED | User Unique number (FOREIGN KEY) |
| acct\_type | VARCHAR(25) | Type of account (saving/checking/creditcard) |
| acct\_number | BIGINT | Account number (PRIMARY KEY) |
| balance | DECIMAL(10,2) | Account balance |
| date\_created | DATETIME | Date created |
| max\_limit | DECIMAL(10,2) | Widrawal limit |

* 1. **Transactions** Table
     1. **Transaction id must be unique**

|  |  |  |
| --- | --- | --- |
| Field | Datatype | Description |
| acct\_number | BIGINT | Account number (FOREIGN KEY) |
| id | BIGINT UNSIGNED | Transaction id (PRIMARY KEY) |
| date | DATETIME | Transaction date |
| type | VARCHAR(50) | Transaction type (grocery/food/utility/retail/etc) |
| amount | DECIMAL(0,2) | Transacted amount |

* 1. **Messages** Table
     1. **Message id must be unique**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| user\_id | INT UNSIGNED | User unique id (FOREIGN KEY) |
| id | INT UNSIGNED | Message Id (PRIMARY KEY) |
| date | DATETIME | Date message composed |
| remark | TEXT | Message detail max 65535 chars |



1. Populate data
   1. Generate dataset using data from [mockaroo](https://www.mockaroo.com/)
      1. Users.csv
      2. Accounts.csv
      3. Transactions.sql
      4. Messages.sql
2. Query to find spending pattern
   1. Monthly average spending for each user

// average monthly transactions for each users

SELECT a.user\_id, a.acct\_number, t.date, AVG(t.amount)

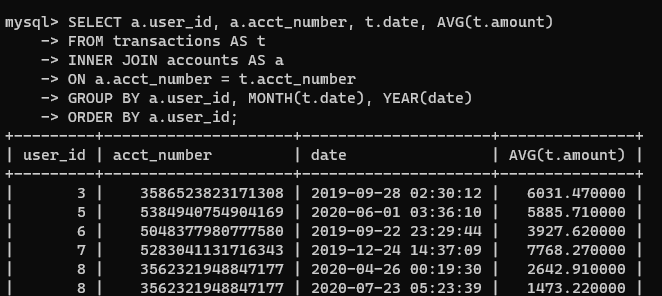
FROM transactions AS t

INNER JOIN accounts AS a

ON a.acct\_number = t.acct\_number

GROUP BY a.user\_id, MONTH(t.date), YEAR(date)

ORDER BY a.user\_id;



* 1. Most frequent transaction type (Issue to display correct type name BUT count is correct.
     1. First query each user same transaction type count
     2. Then, use the new table to find the max of each user type count

SELECT t1.user\_id, t1.ttype, MAX(t1.ctype)

FROM (SELECT a.user\_id, a.acct\_number, t.date, t.type as ttype, COUNT(t.type) as ctype

      FROM transactions AS t

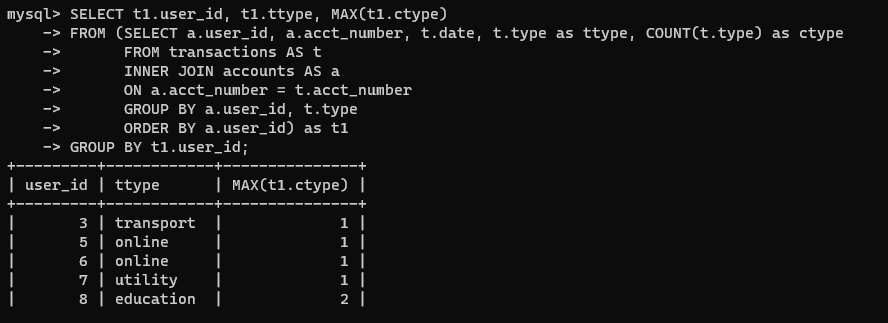
      INNER JOIN accounts AS a

      ON a.acct\_number = t.acct\_number

      GROUP BY a.user\_id, t.type

      ORDER BY a.user\_id) as t1

GROUP BY t1.user\_id;



* 1. Customer who have highest and lowest balance
     1. Highest balance

SELECT u.user\_id, u.name, SUM(a.balance) AS total FROM accounts AS a

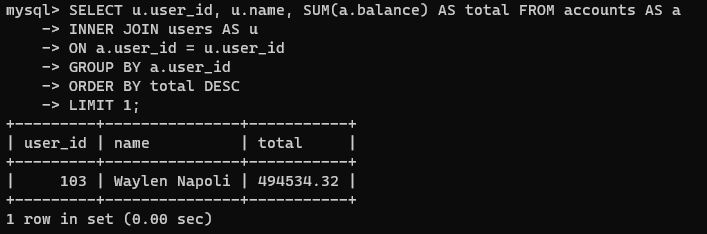
INNER JOIN users AS u

ON a.user\_id = u.user\_id

GROUP BY a.user\_id

ORDER BY total DESC

LIMIT 1;



1. Lowest balance

SELECT u.user\_id, u.name, SUM(a.balance) AS total FROM accounts AS a

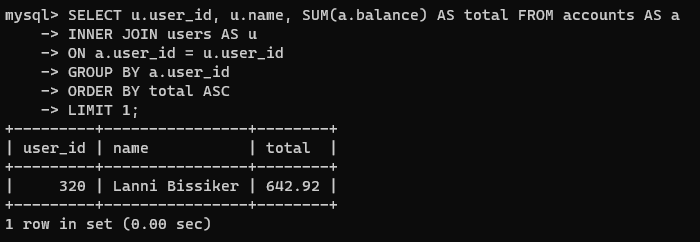
INNER JOIN users AS u

ON a.user\_id = u.user\_id

GROUP BY a.user\_id

ORDER BY total ASC

LIMIT 1;



1. Find user who had the most transaction (all transaction type) for each month

SELECT a.user\_id, u.name, a.acct\_number, t.date, COUNT(a.user\_id) as cmonth

FROM transactions AS t

INNER JOIN accounts AS a

ON a.acct\_number = t.acct\_number

INNER JOIN users AS u

ON u.user\_id = a.user\_id

GROUP BY a.user\_id, MONTH(t.date), YEAR(t.date)

ORDER BY cmonth DESC

LIMIT 1;

