



[DEVS' TOOLKIT PART 2]

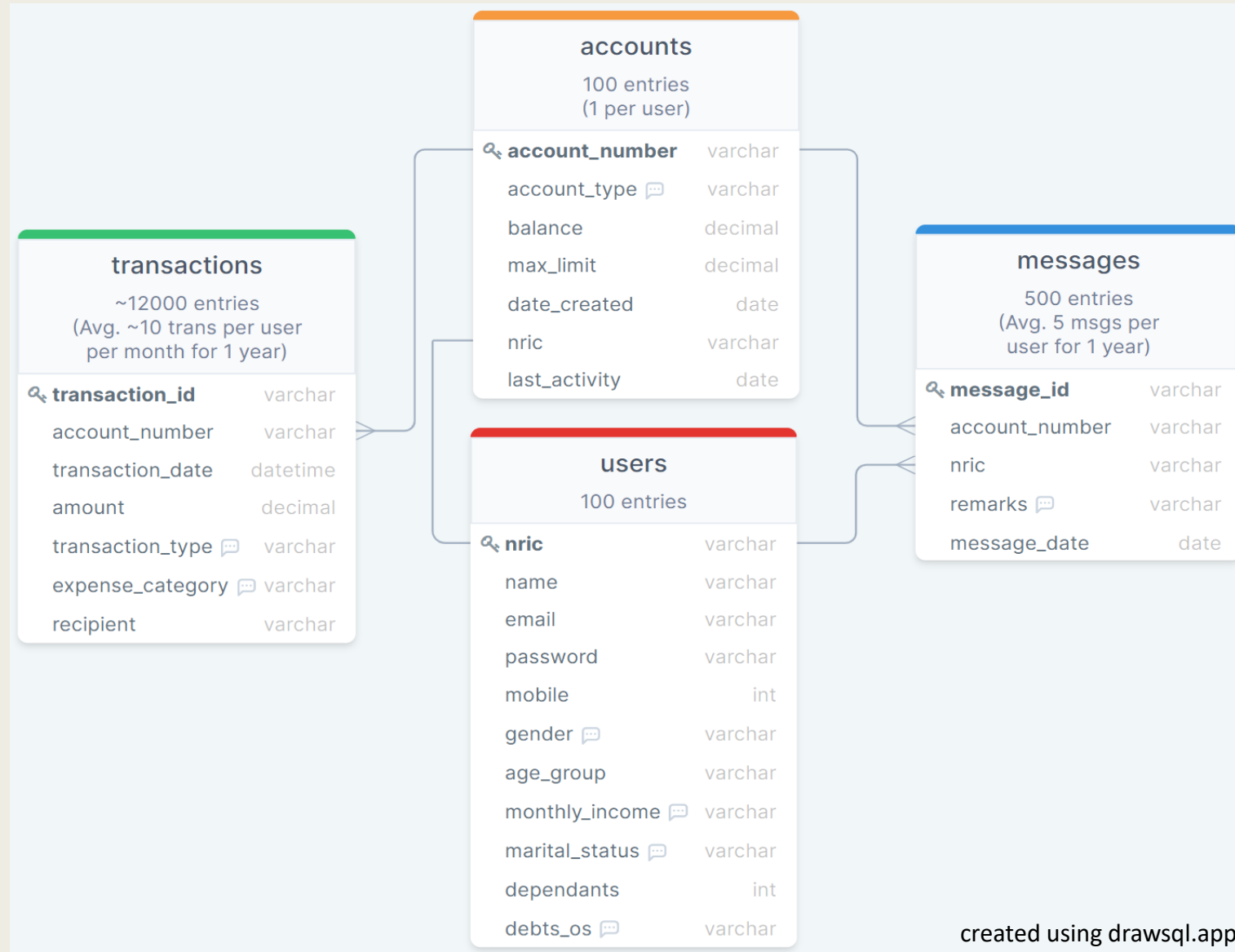
HOMEWORK 1

Database Design for NUSmoney App

by Arvind Subramanian



QN 1: DESIGN DATABASE SCHEMA



QN 2: POPULATE DATASET

- Populated using **Mockaroo Random Data Generator**
- Mix of Pre-Packaged Lists and Distributions
 - Pre-Packaged Lists: names, serial numbers etc.
 - Random Distributions: age group, gender, dates etc.
 - Custom Distributions: income, dependants etc. *(for columns influenced by other columns)*
- Implemented various custom distribution correlations, such as:
 - users.income and users.age_group
 - accounts.balance and users.income / users.age_group
 - users.dependants and transactions.transaction_type
 - Improves simulation realism
- Consistent pairing of NRIC and Account Numbers across tables

After data-loading ----->

[4 .sql files can be found in the zipped folder](#)

```
Select MySQL 8.0 Command Line Client
mysql> show tables;
+-----+
| Tables_in_nusmoneyapp |
+-----+
| accounts               |
| messages               |
| transactions           |
| users                  |
+-----+
4 rows in set (0.01 sec)

mysql> describe users;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| nric       | varchar(9)    | NO   | PRI | NULL    |       |
| name       | varchar(19)   | NO   |     | NULL    |       |
| email      | varchar(32)   | NO   |     | NULL    |       |
| password   | varchar(12)   | NO   |     | NULL    |       |
| mobile     | int           | NO   |     | NULL    |       |
| gender     | varchar(6)    | NO   |     | NULL    |       |
| age_group  | varchar(5)    | NO   |     | NULL    |       |
| monthly_income | varchar(6)    | NO   |     | NULL    |       |
| marital_status | varchar(1)    | NO   |     | NULL    |       |
| dependants | int           | NO   |     | NULL    |       |
| debts_os   | varchar(18)   | NO   |     | NULL    |       |
+-----+
11 rows in set (0.01 sec)

mysql> describe accounts;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| account_number | varchar(11)   | NO   | PRI | NULL    |       |
| nric         | varchar(9)    | NO   | MUL | NULL    |       |
| account_type  | varchar(7)    | NO   |     | NULL    |       |
| balance       | decimal(9,2)  | NO   |     | NULL    |       |
| max_limit     | int           | NO   |     | NULL    |       |
| date_created  | date          | NO   |     | NULL    |       |
| last_activity | date          | NO   |     | NULL    |       |
+-----+
7 rows in set (0.01 sec)

mysql> describe transactions;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| transaction_id | varchar(10)   | NO   | PRI | NULL    |       |
| transaction_date | datetime      | NO   |     | NULL    |       |
| account_number | varchar(11)   | NO   | MUL | NULL    |       |
| transaction_type | varchar(14)   | NO   |     | NULL    |       |
| expense_category | varchar(21)   | NO   |     | NULL    |       |
| amount         | decimal(7,2)  | NO   |     | NULL    |       |
| recipient      | varchar(13)   | NO   |     | NULL    |       |
+-----+
7 rows in set (0.00 sec)

mysql> describe messages;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| message_id | varchar(11)   | NO   | PRI | NULL    |       |
| account_number | varchar(11)   | NO   | MUL | NULL    |       |
| nric       | varchar(9)    | NO   | MUL | NULL    |       |
| remarks    | varchar(20)   | NO   |     | NULL    |       |
| message_date | date          | NO   |     | NULL    |       |
+-----+
5 rows in set (0.00 sec)

mysql>
```

QN 3: QUERY THE DATABASE

Spending Patterns

- Total spent by all customers on each of the expense categories over 1 year

```
select expense_category, sum(amount) as total_spent from transactions group by expense_category order by total_spent desc;
```

- Spending in each category broken down by age group
(This code can be easily modified to visualise the relationship with any other factor in the users table)

```
select t.expense_category, aa.age_group, sum(t.amount) as total_spent
from transactions t left join (
select a.account_number, u.age_group
from users u left join accounts a
on u.nric = a.nric) as aa
on t.account_number = aa.account_number
group by t.expense_category, aa.age_group
order by aa.age_group, total_spent desc;
```

```
Select MySQL 8.0 Command Line Client
mysql>
mysql> select expense_category, sum(amount) as total_spent from transactions group by expense_category order by total_spent desc;
+-----+-----+
| expense_category | total_spent |
+-----+-----+
| investments      | 718580.28   |
| bills & utilities | 314625.88   |
| household & groceries | 295583.69   |
| food             | 287287.19   |
| loans            | 242142.80   |
| entertainment     | 229338.70   |
| insurance         | 204568.86   |
| shopping          | 188262.64   |
| transport         | 180980.95   |
| others            | 107222.90   |
+-----+-----+
10 rows in set (0.01 sec)

mysql> select t.expense_category, aa.age_group, sum(t.amount) as total_spent
-> from transactions t left join (
-> select a.account_number, u.age_group
-> from users u left join accounts a
-> on u.nric = a.nric) as aa
-> on t.account_number = aa.account_number
-> group by t.expense_category, aa.age_group
-> order by aa.age_group, total_spent desc;
+-----+-----+-----+
| expense_category | age_group | total_spent |
+-----+-----+-----+
| investments      | <25       | 267975.36   |
| food             | <25       | 130800.85   |
| shopping          | <25       | 71350.29    |
| entertainment     | <25       | 71350.28    |
| loans            | <25       | 65684.07    |
| insurance         | <25       | 45244.76    |
| transport         | <25       | 42278.49    |
| others            | <25       | 17180.84    |
| bills & utilities | <25       | 11683.91    |
| household & groceries | <25       | 7337.08     |
| bills & utilities | >65       | 123230.92   |
| household & groceries | >65       | 117980.64   |
| insurance         | >65       | 88544.56    |
| investments      | >65       | 70251.46    |
| entertainment     | >65       | 65388.98    |
| others            | >65       | 52727.74    |
| shopping          | >65       | 48842.50    |
| transport         | >65       | 45853.89    |
| food             | >65       | 33160.74    |
| loans            | >65       | 31343.67    |
| investments      | 25-45    | 194098.68   |
| bills & utilities | 25-45    | 100326.79   |
| household & groceries | 25-45    | 95731.54    |
| loans            | 25-45    | 90843.29    |
| food             | 25-45    | 68280.09    |
| transport         | 25-45    | 52694.47    |
| entertainment     | 25-45    | 45563.13    |
| insurance         | 25-45    | 35463.65    |
| shopping          | 25-45    | 32322.25    |
| others            | 25-45    | 20754.08    |
| investments      | 45-65    | 186254.78   |
| bills & utilities | 45-65    | 79384.26    |
| household & groceries | 45-65    | 74534.43    |
| food             | 45-65    | 55125.51    |
| loans            | 45-65    | 54271.77    |
| entertainment     | 45-65    | 47036.31    |
| transport         | 45-65    | 40163.10    |
| shopping          | 45-65    | 35738.60    |
| insurance         | 45-65    | 35615.89    |
| others            | 45-65    | 16560.24    |
+-----+-----+-----+
40 rows in set (0.03 sec)

mysql>
```

QN 3: QUERY THE DATABASE

Monthly Average Spending for Each User

```
select u.nric, u.name, temp.account_number, temp.average_monthly_expenditure
from users u left join (
select a.nric, t.account_number, round(sum(t.amount)/count(distinct
date_format(t.transaction_date, '%m')),2) as average_monthly_expenditure
from transactions t left join accounts a
on t.account_number = a.account_number
group by t.account_number) as temp
on u.nric = temp.nric;
```

MySQL 8.0 Command Line Client

```
mysql>
mysql>
mysql>
mysql> select u.nric, u.name, temp.account_number, temp.average_monthly_expenditure
-> from users u left join (
-> select a.nric, t.account_number, round(sum(t.amount)/count(distinct date_format(t.transaction_date, '%m')),2) as average_monthly_expenditure
-> from transactions t left join accounts a
-> on t.account_number = a.account_number
-> group by t.account_number) as temp
-> on u.nric = temp.nric;
```

nric	name	account_number	average_monthly_expenditure
S0001541U	Malena Awty	130679634-2	2708.25
S0185480N	Lynnell Breadmore	574081326-3	1649.61
S0198169A	Gussy Kinkaid	301210885-8	2272.92
S0374824X	Gaby Medley	307212344-6	2283.20
S0430809M	Johnathon Spacy	878472959-7	2471.06
S0826999K	Adelind Lockhart	642339705-8	2711.95
S0991818P	Joshua Delahunty	102026189-7	1713.28
S0999166E	Nathalia Graeber	822093550-2	2753.78
S1079819V	Avigdor Kerrigan	052559836-7	2515.37
S1141299M	Barnard Kitchen	243033543-3	2306.49
S1154086J	Titos Batt	395482849-9	2184.08
S1595908F	Iormina Gilford	042279062-1	2465.36
S17005910	Erma Belmont	642577676-5	2071.90
S1876474U	Sig Wilder	473933937-4	3005.05
S2126359K	Lilie Decze	464385920-2	1969.51
S2130358Q	Lacie Burnsall	720255969-2	2226.33
S2371070B	Cristobal Hanster	776048087-3	1993.02
S2454608W	Ailey Filewood	038324752-7	2399.46
S2547893W	Egon Urch	188710633-2	1820.63
S2573541Q	Celle Challice	715418917-7	2508.16
S2621949U	Matias Bremmell	036679731-X	2122.71
S2802524G	Edy Stokey	846271253-X	2680.77
S2834440B	Ruperta Goldsack	443403900-8	2187.07
S2834966D	Charla Dudley	133463382-7	2422.22
S3354615Z	Allayne Pirri	008212370-5	2171.90
S3360527Z	Adams Bracegirdle	751721798-4	2369.28
S3365010F	Alphard Farady	929763553-9	2108.96
S3407187T	Janina Sarvar	952178669-8	3107.12
S3622022M	Rollo Pietzner	687798011-5	2417.12
S4244651E	Frasco Caulkett	101385969-3	2525.93
S4299805Z	Audra Laughtisse	025886852-X	2280.13
S44547290	Bryon Welton	657463541-1	2081.67
S4460879K	Maryjane Lehemann	534034340-5	2745.81
S4497387A	Bunni Praten	512922811-1	2359.05
S4863738H	Quent Bafford	715352392-8	2510.31
S4879423Q	Freemon Yakovliv	931180907-1	1754.86
S4953512G	Isabelle Hassey	192450351-4	2795.77
S4953584L	Mellie Hurd	464548741-8	2075.95
S5207216A	Ragnar Bailey	135353089-2	2285.64
S5407228E	Doti Agg	303180529-1	2476.87
S5456493G	Madlin Feenan	770099633-4	1969.54
S5477330I	Lionello Fibbens	104224203-4	2650.42
S5558784K	Oralle Schankel	501992816-X	2080.06
S5745108R	Imogene Semrad	705752338-6	2047.59
S5754733B	Lorens O'Dyvoie	722260612-2	2736.10
S5816445E	Patton Zannuto	131042556-6	2038.25
S8811335Q	Natty Craythorn	871838365-3	1994.43
S9154193E	Teddie Rait	748088466-4	2188.84
S9193451G	Nanete Borgars	264443005-4	1970.33
S9300946B	Luis Cropton	171416925-1	2727.92
S9444925C	Correy Pendrick	989210982-1	1972.66
S9571154F	Charo Ovell	586829246-4	2318.20
S9623048K	Bret Mijis	950497465-1	2465.44
S9652035U	Swen Upex	469971535-4	1421.10
S96654380	Iormina Glenton	797924414-1	2056.51
S9666144W	Ruperta Leadstone	550771667-6	2897.89
S9803856Z	Lani Harsent	496991906-2	2462.57

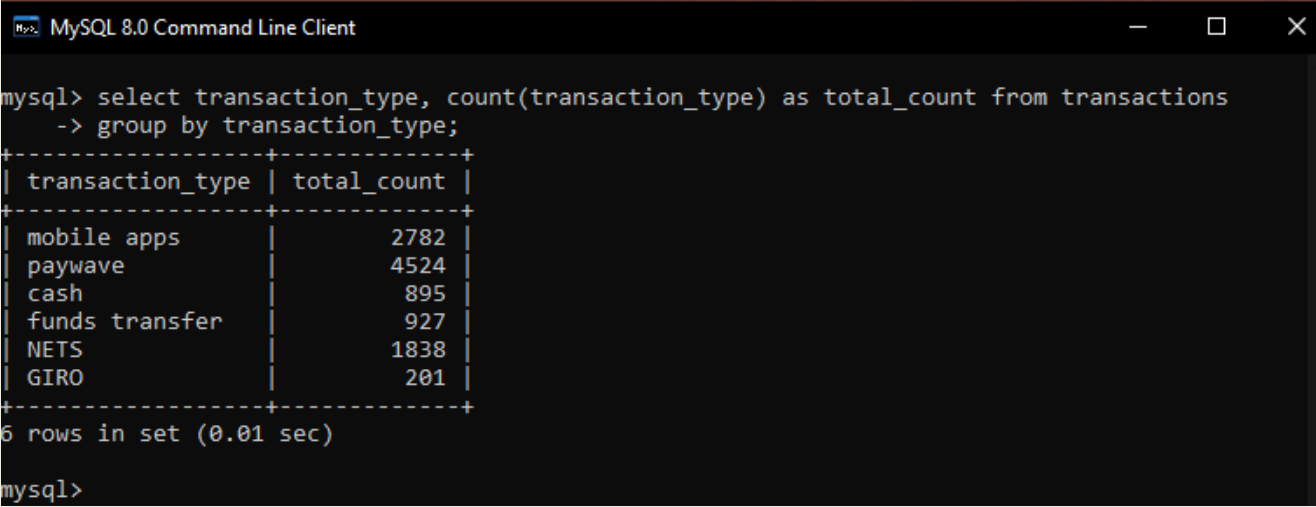
100 rows in set (0.02 sec)

mysql>

QN 3: QUERY THE DATABASE

Most Frequent Transaction Type

```
select transaction_type, count(transaction_type) as total_count from transactions  
group by transaction_type;
```



```
mysql> select transaction_type, count(transaction_type) as total_count from transactions  
-> group by transaction_type;
```

transaction_type	total_count
mobile apps	2782
paywave	4524
cash	895
funds transfer	927
NETS	1838
GIRO	201

```
6 rows in set (0.01 sec)  
  
mysql>
```

QN 3: QUERY THE DATABASE

Information of Customers with the Highest and Lowest Balance

```
select u.nric, u.name, u.mobile, u.email, a.account_number, a.balance, u.gender,
u.age_group, u.monthly_income, u.marital_status, u.dependants, u.debts_os
from users u inner join (
select nric, account_number, balance from accounts
where balance in (select max(balance) from accounts)) as a
on u.nric = a.nric;
```

```
select u.nric, u.name, u.mobile, u.email, a.account_number, a.balance, u.gender,
u.age_group, u.monthly_income, u.marital_status, u.dependants, u.debts_os
from users u inner join (
select nric, account_number, balance from accounts
where balance in (select min(balance) from accounts)) as a
on u.nric = a.nric;
```

```
mysql> select u.nric, u.name, u.mobile, u.email, a.account_number, a.balance, u.gender, u.age_group, u.monthly_income, u.marital_status, u.dependants, u.debts_os
-> from users u inner join (
-> select nric, account_number, balance from accounts
-> where balance in (select max(balance) from accounts)) as a
-> on u.nric = a.nric;
```

nric	name	mobile	email	account_number	balance	gender	age_group	monthly_income	marital_status	dependants	debts_os
S5407228E	Doti Agg	96349646	dagg28@bbb.org	303180529-1	199671.23	Female	45-65	10-15k	M	1	>12 x monthly pay

```
1 row in set (0.00 sec)
```

```
mysql> select u.nric, u.name, u.mobile, u.email, a.account_number, a.balance, u.gender, u.age_group, u.monthly_income, u.marital_status, u.dependants, u.debts_os
-> from users u inner join (
-> select nric, account_number, balance from accounts
-> where balance in (select min(balance) from accounts)) as a
-> on u.nric = a.nric;
```

nric	name	mobile	email	account_number	balance	gender	age_group	monthly_income	marital_status	dependants	debts_os
S9154193E	Teddie Rait	96388039	trait2g@miibeian.gov.cn	748088466-4	1232.01	Female	<25	<5k	S	0	0

```
1 row in set (0.00 sec)
```

```
mysql>
```

QN 4: LAST QUERY

Users who had the most transactions (all transaction types) for each month

```
select ttt.month, ttt.most_trans, u.nric, u.name, ttt.account_number, u.mobile
from users u inner join (
select tt.month, a.nric, tt.account_number, tt.most_trans
from accounts a inner join (
select t.mths as month, t.account_number, max(t.tots) as most_trans
from (select date_format(transaction_date, '%m') as mths, account_number, count(amount) as
tots from transactions group by mths, account_number order by mths asc, tots desc) as t
-> group by t.mths) as tt
-> on tt.account_number = a.account_number) as ttt
-> on ttt.nric = u.nric
-> order by ttt.month asc;
```

```
Select MySQL 8.0 Command Line Client
mysql> select ttt.month, ttt.most_trans, u.nric, u.name, ttt.account_number, u.mobile
-> from users u inner join (
-> select tt.month, a.nric, tt.account_number, tt.most_trans
-> from accounts a inner join (
-> select t.mths as month, t.account_number, max(t.tots) as most_trans
-> from (select date_format(transaction_date, '%m') as mths, account_number, count(amount) as
tots from transactions group by mths, account_number order by mths asc, tots desc) as t
-> group by t.mths) as tt
-> on tt.account_number = a.account_number) as ttt
-> on ttt.nric = u.nric
-> order by ttt.month asc;
```

month	most_trans	nric	name	account_number	mobile
01	17	S0999166E	Nathalia Graeber	822093550-2	95790876
02	17	S8068215M	Salvidor Trendle	249388807-3	95053724
03	18	S1079819V	Avigdor Kerrigan	052559836-7	96622864
04	16	S6892188N	Krishnah Keefe	745917395-9	91346761
05	17	S6652534J	Pail Davinet	553347372-6	91777299
06	18	S96654380	Iormina Glenton	797924414-1	91246001
07	20	S0826999K	Adelind Lockhart	642339705-8	96033768
08	21	S0430809M	Johnathon Spacy	878472959-7	90232383
09	18	S0198169A	Gussy Kinkaid	301210885-8	93865422
10	15	S6682048Z	Lenore Gaw	776335981-1	96821838
11	18	S4244651E	Frasco Caulkett	101385969-3	91493445
12	16	S0430809M	Johnathon Spacy	878472959-7	90232383

```
12 rows in set (0.01 sec)

mysql>
```




END