Database Design

Tables

- Cards
- Cards Bonus Benefits
- User Profile
- User owned cards
- User spendinigs

Card_Catalogue

Column Name	Primary Key / Foreign Key	Constraint	Data Type
Card_ID	Primary key	unique, not null	integer
Bank		not null	enum {1: DBS}, 2: Citi
Card_Name		not null	string
Benefit_type		not null	enum {1: Miles, 2: Cashback, 3: Both}
base_benefit_rate (in $ per $1 spent)		not null	decimal
status		not null	enum(valid, invalid)

Table Constraints

1. **Primary key**

- `PRIMARY KEY (Card_ID)`

2. **Uniqueness to prevent duplicate catalogue entries**

- `UNIQUE (Bank, Card_Name)`
  - prevents you from seeding the same card name twice for the same bank.

3. **Base rate must be non-negative**

- CHECK (base_benefit_rate >= 0)

Card Bonus Categories - Separate table as one card can have separate bonus categories with separate caps. Base rate (e.g. 0.3%) can remain in the cards table because each card only has one base rate. For unlimited cards with cap cashback (e.g. after $60 then will earn 0.3% instead of 1.5%), there is a category called All. If All, and within bonus cashback cap, then any item the user sets on the input page will be eligible for bonus (1.5%).

Added Status column: For cards that are no longer available, status will be inactive. We will not delete because users can have tracked spendings under the card, and users will not want to suddenly see all their spendings under certain cards get deleted. To show the user spending under each card in the app, it joins user_spending table to this cards table via card_id.

Column Name	Primary Key / Foreign Key	Constraint	Data Type
Card_BonusCat_ID	Primary key	unique, not null	integer
Card_ID	FK (to cards table)	not null	integer
bonus_category		not null	enum{1: Food, 2: Transport, 3: Entertainment, 4: Fashion, 5: All)
bonus_benefit_rate (in $ per $1 spent)		not null	decimal
bonus_cap_in_dollar		not null, default 9999999	integer
bonus_minimum_spend_in_dollar		not null, default 0	integer

Table Level Constraints

1. **Primary key**

- `PRIMARY KEY (Card_BonusCat_ID)`

2. **Foreign key to Cards**

- `FOREIGN KEY (Card_ID) REFERENCES Cards(Card_ID)`
  - (and if you want cleanup behaviour: `ON DELETE CASCADE` so deleting a card deletes its bonus rules)

3. **Prevents duplicate category rules per card**

- `UNIQUE (Card_ID, bonus_category)`

Card Selections

Standard Chartered Simply Cash, Citi PremierMiles, DBS Live Fresh

```
INSERT INTO Cards (Card_ID, Bank, Card_Name, Benefit_type, base_benefit_rate)
VALUES
  -- Unlimited cashback: 1.5% cashback ⇒ 0.015 per $1 spent
  (1, 'StandardChartered', 'Simply Cash (Unlimited Cashback)', 'Cashback', 0.015000),

  -- Miles card: 1.2 miles per S$1 local spend
  (2, 'Citi', 'Citi PremierMiles', 'Miles', 1.200000),

  -- Cashback card with higher headline cashback than 1.5%: base 0.3% ⇒ 0.003 per $1
  (3, 'DBS', 'DBS Live Fresh', 'Cashback', 0.003000);

INSERT INTO Card_Bonus_Categories
  (Card_BonusCat_ID, Card_ID, bonus_category, bonus_benefit_rate, bonus_cap_in_dollar, bonus_minimum_spend_in_dollar)
VALUES
  -- DBS Live Fresh: additional 5.7% on Shopping (mapped to Fashion), capped at $50, min spend $800
  (301, 3, 'Fashion',   0.057000, 50.00, 800.00),

  -- DBS Live Fresh: additional 5.7% on Transport, capped at $20, min spend $800
  (302, 3, 'Transport', 0.057000, 20.00, 800.00);
```

User Profile

Column Name	PK / FK	Constraint	Data Type
User_ID	PK	unique, not null	integer
Name			string
Preference			enum {Miles, Cashback, Both}

username			VARCHAR(50)
Email			VARCHAR(100)
password_hash			VARCHAR(255)
status			ENUM('active','inactive','suspended')
created_at			TIMESTAMP / DATETIME
last_modified_at			TIMESTAMP

User Owned Cards

Column Name	PK/FK	Constraint	Data Type
User_Card_ID	PK	unique, not null	integer
User_ID	FK (to user profile table)	not null	integer
Card_ID	FK (to cards table)	not null	integer
Card_expiry_date		not null, default 9999-12-31	date
Date_of_monthly_cycle_refresh		not null, DEFAULT (LAST_DAY(CURDATE()))	date
Status		not null	enum {Active, Suspended, Expired}

User Spending

Column Name	PK/FK	Constraint	Data Type
user_spending_id	PK	unique, not null	integer

User_ID	FK (to user profile table)	not null	integer
Card_ID	FK (to card table)		integer
spend_date		not null, default getdate	date
Spend_amt		not null	Decimal (in SQLAlchemy is Numeric(10,2) e.g.)
Category		not null	Enum {Food, Transport, Entertainment, Fashion,...}

- Conversion rate between miles and dollar (think can just be a variable, instead of having a table for it)

Audit Log

Column Name	PK/FK	Constraint	Data Type
audit_id	PK	unique, not null	integer

```
CREATE TABLE audit_log (
  audit_id BIGINT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NULL,
  event_type ENUM(
    'CARD_ADDED','CARD_UPDATED','CARD_DELETED',
    'SPEND_ADDED','SPEND_DELETED',
    'RECO_RUN','GENAI_TRIGGERED'
  ) NOT NULL,
  entity_table VARCHAR(64) NULL,
  entity_id BIGINT NULL,
  event_ts DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  metadata_json JSON NULL,
  INDEX ix_audit_user_ts (user_id, event_ts),
  CONSTRAINT fk_audit_user FOREIGN KEY (user_id) REFERENCES user_profile(user_id)
```

```
    ON DELETE SET NULL
);
```