Mandiri Sekuritas — Data Analyst Technical Test

Project: mandiri-sekuritas-469605
Dataset: MandiriSekuritas
Tables:

- MandiriSekuritas.Transaction
- MandiriSekuritas.Users
- MandiriSekuritas.Cards

Time scope analyzed: **2010–2019** (UTC converted to **Asia/Jakarta**)

---

1) Goal

Produce reusable SQL views and exports to analyze **user transaction behavior**:

- Overview KPIs (txn count, value, avg ticket, active users)
- Trend over time (daily)
- Channel usage (Chip vs Swipe vs Unknown)
- Merchant categories (MCC)
- Segmentation by Age band

These outputs are used to build a **Looker Studio** dashboard.

2) Requirements

- Google BigQuery access to project mandiri-sekuritas-469605
- Permissions: BigQuery Job User + BigQuery Data Viewer (and BigQuery Data Owner if you will create views/tables)

---

3) Data & Key Assumptions

- All timestamps in Transaction.date are converted to **Asia/Jakarta** using BigQuery's DATETIME(timestamp, "Asia/Jakarta").
- **Average Ticket** = SUM(amount) / COUNT(*).
- **Active Users** = COUNT(DISTINCT client_id) for the aggregation period.

---

4) How to Run

Create reusable views

Copy each query below and wrap it with CREATE OR REPLACE VIEW. Example:

```
CREATE OR REPLACE VIEW
`mandiri-sekuritas-469605.MandiriSekuritas.kpi_overview` AS
-- paste the full Overview KPIs query body here (starting from WITH base AS ...)
```

Do the same for each section:

- kpi_overview
- trend_over_time
- channel_usage
- mcc_usage
- segmentation_age

---

5) SQL — Queries (ready for View creation)

5.1 Overview KPIs — quantify activity & scale

Purpose: Yearly KPI snapshot (uses Jakarta local date).

```
WITH base AS (
  SELECT
    t.id AS txn_id,
    t.date AS ts_utc,
    DATETIME(t.date, "Asia/Jakarta") AS ts_jkt,
    DATE(DATETIME(t.date, "Asia/Jakarta")) AS trade_date,
    EXTRACT(YEAR  FROM DATETIME(t.date, "Asia/Jakarta")) AS yr,
    EXTRACT(MONTH FROM DATETIME(t.date, "Asia/Jakarta")) AS mo,
    EXTRACT(DAY   FROM DATETIME(t.date, "Asia/Jakarta")) AS dd,
    EXTRACT(HOUR  FROM DATETIME(t.date, "Asia/Jakarta")) AS hh,
    t.client_id, t.card_id, t.amount,
    CASE
      WHEN LOWER(t.use_chip) LIKE '%chip%'  THEN 'Chip'
      WHEN LOWER(t.use_chip) LIKE '%swipe%' THEN 'Swipe'
      ELSE 'Unknown'
    END AS channel,
    t.merchant_id, t.merchant_city, t.merchant_state, t.zip, t.mcc, t.errors,
    t.amount < 0 AS is_refund
  FROM `mandiri-sekuritas-469605.MandiriSekuritas.Transaction` t
),
fact AS (
  SELECT
    b.*,
    u.current_age, u.retirement_age, u.birth_year, u.birth_month, u.gender,
    u.per_capita_income, u.yearly_income, u.total_debt, u.credit_score,
u.num_credit_cards,
    c.card_brand, c.card_type, c.has_chip, c.credit_limit, c.acct_open_date,
c.card_on_dark_web,
    CASE
      WHEN u.current_age IS NULL THEN 'Unknown'
      WHEN u.current_age < 20               THEN '<20'
      WHEN u.current_age BETWEEN 20 AND 24    THEN '20-24'
      WHEN u.current_age BETWEEN 25 AND 29    THEN '25-29'
      WHEN u.current_age BETWEEN 30 AND 34    THEN '30-34'
      WHEN u.current_age BETWEEN 35 AND 39    THEN '35-39'
      WHEN u.current_age BETWEEN 40 AND 49    THEN '40-49'
      WHEN u.current_age BETWEEN 50 AND 59    THEN '50-59'
      ELSE '60+'
    END AS age_band,
    CASE
      WHEN u.yearly_income IS NULL THEN 'Unknown'
      WHEN u.yearly_income < 20000 THEN '<20k'
      WHEN u.yearly_income < 40000 THEN '20k-40k'
      WHEN u.yearly_income < 60000 THEN '40k-60k'
      WHEN u.yearly_income < 80000 THEN '60k-80k'
      ELSE '80k+'
    END AS income_band,
    CASE
```

```
      WHEN u.credit_score IS NULL THEN 'Unknown'
      WHEN u.credit_score < 600   THEN '<600'
      WHEN u.credit_score < 700   THEN '600-699'
      WHEN u.credit_score < 750   THEN '700-749'
      ELSE '750+'
    END AS credit_band,
    (b.channel = 'Swipe' AND c.has_chip = TRUE) AS chip_capable_but_swiped,
    (c.card_on_dark_web = TRUE) AS dark_web_flag
  FROM base b
  LEFT JOIN `mandiri-sekuritas-469605.MandiriSekuritas.Users` u ON b.client_id =
u.id
  LEFT JOIN `mandiri-sekuritas-469605.MandiriSekuritas.Cards` c ON b.card_id = c.id
)
SELECT
  FORMAT_DATE("%Y", trade_date) AS year,  -- string "YYYY"
  COUNT(*)                          AS txn_count,
  SUM(amount)                          AS total_value,
  SAFE_DIVIDE(SUM(amount), COUNT(*))        AS avg_ticket,
  COUNT(DISTINCT client_id)               AS active_users
FROM fact
GROUP BY year
ORDER BY year;
```

5.2 Trend Over Time — daily txn & value)

```
WITH base AS (...same as 5.1...), fact AS (...same as 5.1...)
SELECT
  trade_date,
  COUNT(*)      AS txn_count,
  SUM(amount)   AS total_value
FROM fact
GROUP BY trade_date
ORDER BY trade_date;
```

5.3 Channel Usage — Chip vs Swipe (adoption & risk proxy)

```
WITH base AS (...same as 5.1...), fact AS (...same as 5.1...)
SELECT
  channel,
  COUNT(*)    AS txn_count,
  SUM(amount) AS total_value
FROM fact
GROUP BY channel
ORDER BY txn_count DESC;
```

5.4 Merchant Categories (MCC)

```
WITH base AS (...same as 5.1...), fact AS (...same as 5.1...)
SELECT
  mcc,
  SUM(amount) AS total_value,
  COUNT(*)    AS txn_count
FROM fact
GROUP BY mcc
ORDER BY total_value DESC;
```

5.5 Segmentation by Age

```
WITH base AS (...same as 5.1...), fact AS (...same as 5.1...)
SELECT
  age_band,
  COUNT(*)                                AS txn_count,
  SUM(amount)                             AS total_value,
  SAFE_DIVIDE(SUM(amount), COUNT(*))      AS avg_ticket,
  COUNT(DISTINCT client_id)               AS active_users
FROM fact
GROUP BY age_band
ORDER BY age_band;
```

6) Connect to Looker Studio

1. **Add data source** → BigQuery → choose the **views** created (e.g., kpi_overview, trend_over_time, etc.).
2. For time-series charts, set date field to:
    - trade_date from trend_over_time, or
    - year from kpi_overview (treat as text for a discrete x-axis or convert to date with PARSE_DATE('%Y', year) if needed).
3. **Calculated fields**
    - avg_ticket = total_value / txn_count
    - For monthly labels, create:
        - month_year_label = FORMAT_DATE('%b %Y', trade_date)
        - month_year_sort = CAST(FORMAT_DATE('%Y%m', trade_date) AS NUMBER) (sort dimension)
4. Build visuals:
    - KPI scorecards from kpi_overview
    - Trend lines from trend_over_time
    - Channel bars from channel_usage
    - City bars from merchant
    - Age bars from segmentation_age