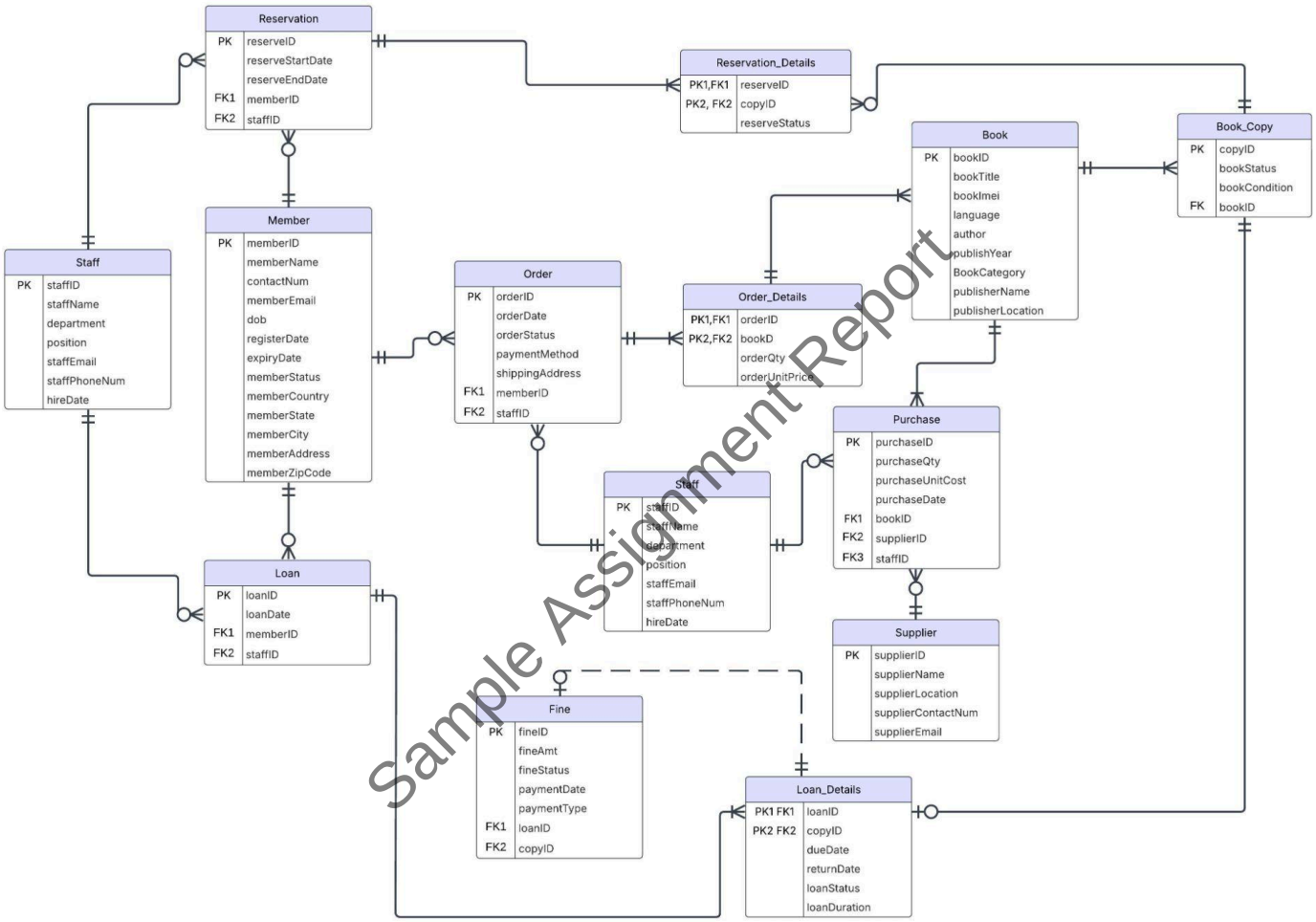
Chapter 1 : Design of Data Warehouse

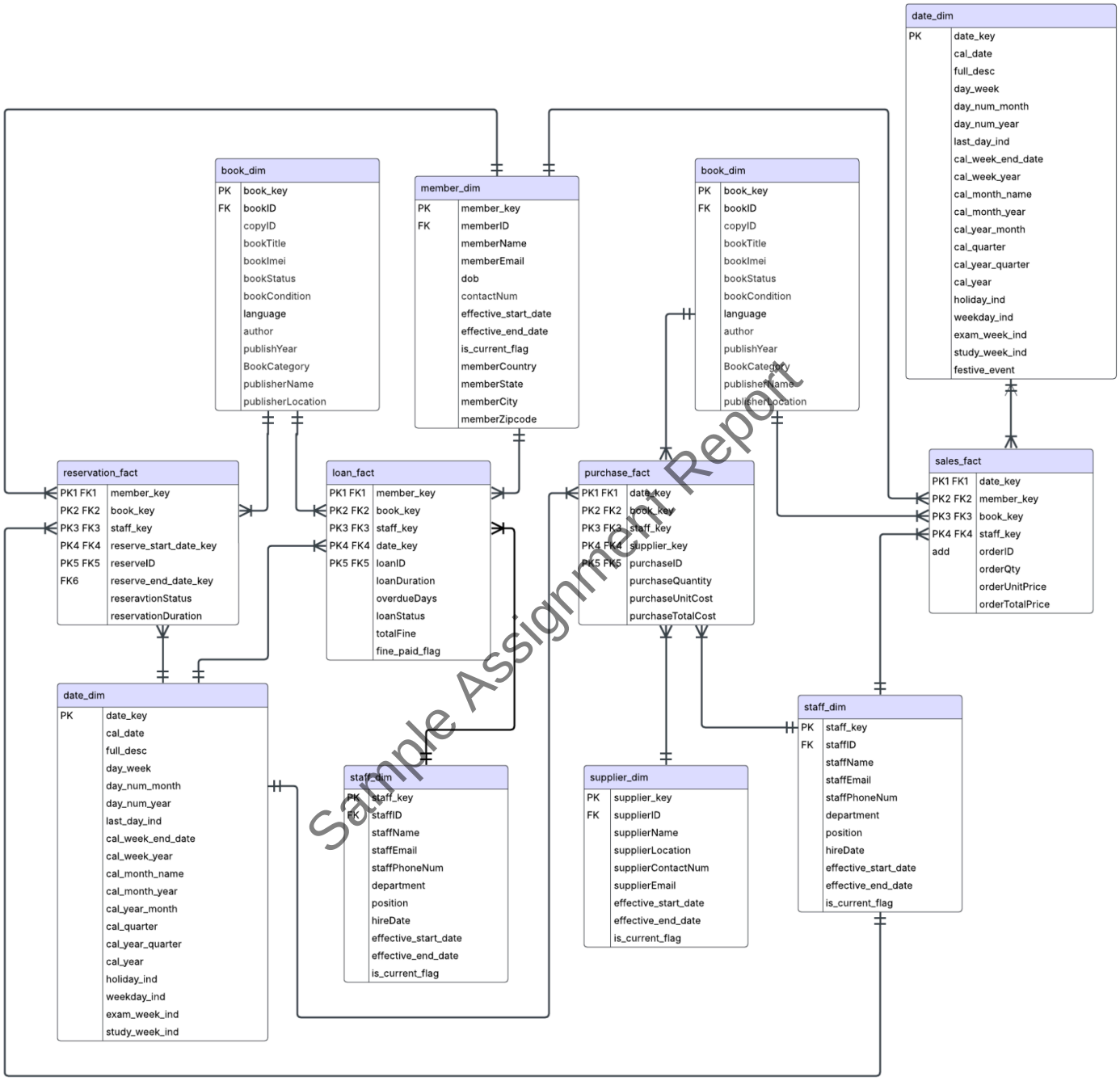
1.1 Logical Design

1.1.1 Original Database



Sample Assignment Report

1.1.2 Star Schema Dimension and Fact Tables



Sample Assignment Report

1.2 Physical Design

1.2.1 Dimension Tables

BOOK_DIM

```
CREATE TABLE book_dim
(
    book_key            NUMBER        NOT NULL,
    bookID              VARCHAR2(6)   NOT NULL,
    copyID              VARCHAR2(7)   NOT NULL,
    bookTitle           VARCHAR2(200) NOT NULL,
    bookImei            VARCHAR2(50),
    language            VARCHAR2(50)  NOT NULL,
    author              VARCHAR2(100) NOT NULL,
    publishYear         NUMBER(4)     NOT NULL,
    bookCategory        VARCHAR2(50)  NOT NULL,
    publisherName       VARCHAR2(100) NOT NULL,
    publisherLocation   VARCHAR2(100) NOT NULL,
    bookStatus          VARCHAR2(20)  NOT NULL,
    bookCondition       VARCHAR2(20)  NOT NULL,
    CONSTRAINT book_dim_pk PRIMARY KEY(book_key),
    CONSTRAINT book_dim_bookid_fk FOREIGN KEY (bookID) REFERENCES BOOK (BOOKID)
);
```

Sample Assignment Report

MEMBER_DIM

```
CREATE TABLE member_dim
(
    member_key           NUMBER          NOT NULL,
    memberID             VARCHAR2(6)     NOT NULL,
    memberName           VARCHAR2(100)   NOT NULL,
    memberEmail          VARCHAR2(100)   NOT NULL,
    dob                  DATE            NOT NULL,
    contactNum           VARCHAR2(20)    NOT NULL,
    memberCountry        VARCHAR2(50)    NOT NULL,
    memberState          VARCHAR2(50)    NOT NULL,
    memberCity           VARCHAR2(50)    NOT NULL,
    memberZipcode        VARCHAR2(10)    NOT NULL,
    effective_start_date DATE            NOT NULL,
    effective_end_date   DATE            NOT NULL,
    is_current_flag      CHAR(1)         NOT NULL,
    CONSTRAINT member_dim_pk PRIMARY KEY (member_key),
    CONSTRAINT member_dim_memberid_fk FOREIGN KEY (memberID) REFERENCES MEMBER
(MEMBERID)
);
```

DATE_DIM

```
CREATE TABLE DATE_DIM
(
    DATE_KEY          NUMBER       NOT NULL,
    CAL_DATE          DATE         NOT NULL,
    FULL_DESC         VARCHAR2(50),
    DAY_WEEK          VARCHAR2(10),
    DAY_NUM_MONTH     NUMBER(2),
    DAY_NUM_YEAR      NUMBER(3),
    LAST_DAY_IND      CHAR(1),
    CAL_WEEK_END_DATE DATE,
    CAL_WEEK_YEAR     NUMBER(2),
    CAL_MONTH_NAME    VARCHAR2(15),
    CAL_MONTH_YEAR    NUMBER(2),
    CAL_YEAR_MONTH    CHAR(7),
    CAL_QUARTER       CHAR(2),
    CAL_YEAR_QUARTER CHAR(7),
    CAL_YEAR          NUMBER(4),
    HOLIDAY_IND       CHAR(1),
    WEEKDAY_IND       CHAR(1),
    EXAM_WEEK_IND     CHAR(1),
    STUDY_WEEK_IND    CHAR(1),
    FESTIVE_EVENT     VARCHAR2(50), -- Added to store holiday names
    CONSTRAINT DATE_DIM_PK PRIMARY KEY(DATE_KEY)
);
```

STAFF_DIM

```
CREATE TABLE staff_dim
(
    staff_key              NUMBER       NOT NULL,
    staffID                VARCHAR2(5)  NOT NULL,
    staffName              VARCHAR2(100) NOT NULL,
    staffEmail             VARCHAR2(100) NOT NULL,
    staffPhoneNum          VARCHAR2(20) NOT NULL,
    department             VARCHAR2(50) NOT NULL,
    position               VARCHAR2(50) NOT NULL,
    hireDate               DATE         NOT NULL,
    effective_start_date   DATE         NOT NULL,
    effective_end_date     DATE         NOT NULL,
    is_current_flag        CHAR(1)      NOT NULL,
    CONSTRAINT staff_dim_pk PRIMARY KEY (staff_key),
    CONSTRAINT staff_dim_staffid_fk FOREIGN KEY (staffID) REFERENCES STAFF
(STAFFID)
);
```

SUPPLIER_DIM

```
CREATE TABLE supplier_dim
(
    supplier_key           NUMBER       NOT NULL,
```

Sample Assignment Report

```
    supplierID              VARCHAR2(5)   NOT NULL,
    supplierName            VARCHAR2(100) NOT NULL,
    supplierLocation        VARCHAR2(100) NOT NULL,
    supplierContactNum      VARCHAR2(20)  NOT NULL,
    supplierEmail           VARCHAR2(100) NOT NULL,
    effective_start_date    DATE          NOT NULL,
    effective_end_date      DATE          NOT NULL,
    is_current_flag         CHAR(1)       NOT NULL,
    CONSTRAINT supplier_dim_pk PRIMARY KEY (supplier_key),
    CONSTRAINT supplier_dim_supplierid_fk FOREIGN KEY (supplierID) REFERENCES
SUPPLIER (SUPPLIERID)
);
```

1.2.1 Fact Tables

RESERVATION_FACT

```
CREATE TABLE RESERVATION_FACT
(
    member_key                NUMBER NOT NULL,
    book_key                  NUMBER NOT NULL,
    staff_key                 NUMBER NOT NULL,
    reserve_start_date_key    NUMBER NOT NULL,
    reserve_end_date_key      NUMBER NOT NULL,
    reserveID                 VARCHAR2(7) NOT NULL,
    reservationStatus         VARCHAR2(20) NOT NULL,
    reservationDuration       NUMBER, -- calculated days between start and end
    CONSTRAINT RF_MEMBER_FK       FOREIGN KEY(member_key) REFERENCES
MEMBER_DIM(member_key),
    CONSTRAINT RF_BOOK_FK         FOREIGN KEY(book_key)   REFERENCES
BOOK_DIM(book_key),
    CONSTRAINT RF_STAFF_FK        FOREIGN KEY(staff_key)  REFERENCES
STAFF_DIM(staff_key),
    CONSTRAINT RF_START_DATE_FK   FOREIGN KEY(reserve_start_date_key)
REFERENCES DATE_DIM(date_key),
    CONSTRAINT RF_END_DATE_FK     FOREIGN KEY(reserve_end_date_key)
REFERENCES DATE_DIM(date_key),
    CONSTRAINT RF_RESERVEID_FK    FOREIGN KEY(reserveID)  REFERENCES
Reservation(reserveID),
    CONSTRAINT RESERVATION_FACT_PK  PRIMARY KEY(member_key, book_key, staff_key,
reserve_start_date_key, reserveID)
);
```

Sample Assignment Report

LOAN_FACT

```
CREATE TABLE LOAN_FACT
(
    member_key     NUMBER NOT NULL,
    book_key       NUMBER NOT NULL,
```

```
    staff_key      NUMBER NOT NULL,
    date_key       NUMBER NOT NULL,
    loanID         VARCHAR2(8) NOT NULL,
    loanDuration   NUMBER NOT NULL,
    overdueDays    NUMBER,
    loanStatus     VARCHAR2(20) NOT NULL,
    totalFine      NUMBER(8,2),
    fine_paid_flag NUMBER(1),
    CONSTRAINT LF_MEMBER_FK FOREIGN KEY(member_key) REFERENCES
MEMBER_DIM(member_key),
    CONSTRAINT LF_BOOK_FK   FOREIGN KEY(book_key)   REFERENCES
BOOK_DIM(book_key),
    CONSTRAINT LF_STAFF_FK  FOREIGN KEY(staff_key)  REFERENCES
STAFF_DIM(staff_key),
    CONSTRAINT LF_DATE_FK   FOREIGN KEY(date_key)   REFERENCES
DATE_DIM(date_key),
    CONSTRAINT LF_LOANID_FK FOREIGN KEY(loanID)     REFERENCES Loan(loanID),
    CONSTRAINT LOAN_FACT_PK PRIMARY KEY(member_key, book_key, staff_key,
date_key, loanID)
);
```

PURCHASE_FACT

```
CREATE TABLE PURCHASE_FACT
(
    date_key            NUMBER NOT NULL,
    book_key            NUMBER NOT NULL,
    staff_key           NUMBER NOT NULL,
    supplier_key        NUMBER NOT NULL,
    purchaseID          VARCHAR2(10) NOT NULL,
    purchaseQuantity    NUMBER NOT NULL,
    purchaseUnitCost    NUMBER(10,2) NOT NULL,
    purchaseTotalCost   NUMBER(12,2) NOT NULL,
    CONSTRAINT PF_DATE_FK     FOREIGN KEY(date_key)     REFERENCES
DATE_DIM(date_key),
    CONSTRAINT PF_BOOK_FK     FOREIGN KEY(book_key)     REFERENCES
BOOK_DIM(book_key),
    CONSTRAINT PF_STAFF_FK    FOREIGN KEY(staff_key)    REFERENCES
STAFF_DIM(staff_key),
    CONSTRAINT PF_SUPPLIER_FK FOREIGN KEY(supplier_key) REFERENCES
SUPPLIER_DIM(supplier_key),
    CONSTRAINT PF_PURCHASEID_FK FOREIGN KEY(purchaseID) REFERENCES
PURCHASE(purchaseID),
    CONSTRAINT PURCHASE_FACT_PK PRIMARY KEY(date_key, book_key, staff_key,
supplier_key, purchaseID)
);
```

SALES_FACT

```
CREATE TABLE SALES_FACT
```

Sample Assignment Report

```
(
    date_key         NUMBER       NOT NULL,
    member_key       NUMBER       NOT NULL,
    book_key         NUMBER       NOT NULL,
    staff_key        NUMBER       NOT NULL,
    orderID          VARCHAR2(7)  NOT NULL,
    orderQty         NUMBER       NOT NULL,
    orderUnitPrice   NUMBER(10,2) NOT NULL,
    orderTotalPrice  NUMBER(12,2) NOT NULL,
    CONSTRAINT SF_DATE_FK   FOREIGN KEY(date_key)   REFERENCES
DATE_DIM(date_key),
    CONSTRAINT SF_MEMBER_FK FOREIGN KEY(member_key) REFERENCES
MEMBER_DIM(member_key),
    CONSTRAINT SF_BOOK_FK   FOREIGN KEY(book_key)   REFERENCES
BOOK_DIM(book_key),
    CONSTRAINT SF_STAFF_FK  FOREIGN KEY(staff_key)  REFERENCES
STAFF_DIM(staff_key),
    CONSTRAINT SF_ORDERID_FK FOREIGN KEY(orderID)   REFERENCES ORDERS(orderID),
    CONSTRAINT SALES_FACT_PK PRIMARY KEY(date_key, member_key, book_key,
staff_key, orderID)
);
```

Sample Assignment Report

Chapter 2: Extract, Transform, Load, and Process

2.1 Script for initial loading

BOOK_DIM

```
-- ====================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ====================================================================
CREATE OR REPLACE VIEW BOOK_STAGING_V AS
SELECT
    b.bookID,
    c.copyID,
    -- TECHNIQUE: Data Cleansing
    CASE
        WHEN TRIM(UPPER(b.bookTitle)) IS NULL OR LENGTH(TRIM(b.bookTitle)) < 2
        THEN 'UNKNOWN TITLE'
        ELSE TRIM(UPPER(b.bookTitle))
    END as clean_bookTitle,
    CASE
        WHEN b.bookImei IS NULL OR TRIM(b.bookImei) = ''
        THEN 'NO-IMEI-' || b.bookID
        ELSE TRIM(b.bookImei)
    END as clean_bookImei,
    CASE
        WHEN UPPER(TRIM(b.language)) IN ('EN', 'ENG', 'ENGLISH') THEN 'ENGLISH'
        WHEN UPPER(TRIM(b.language)) IN ('FR', 'FRA', 'FRENCH') THEN 'FRENCH'
        WHEN UPPER(TRIM(b.language)) IN ('ES', 'SPA', 'SPANISH') THEN 'SPANISH'
        WHEN TRIM(b.language) IS NULL THEN 'UNKNOWN'
        ELSE UPPER(TRIM(b.language))
    END as clean_language,
    CASE
        WHEN TRIM(UPPER(b.author)) IS NULL OR LENGTH(TRIM(b.author)) < 2
        THEN 'UNKNOWN AUTHOR'
        ELSE TRIM(UPPER(b.author))
    END as clean_author,
    CASE
        WHEN b.publishYear IS NULL OR b.publishYear < 1800 OR b.publishYear >
EXTRACT(YEAR FROM SYSDATE)
        THEN 1900  -- Default for invalid years
        ELSE b.publishYear
    END as clean_publishYear,
    CASE
        WHEN UPPER(TRIM(b.category)) IN ('FICTION', 'NOVEL') THEN 'FICTION'
        WHEN UPPER(TRIM(b.category)) IN ('NON-FICTION', 'NONFICTION') THEN
'NON-FICTION'
        WHEN UPPER(TRIM(b.category)) IN ('SCIENCE', 'SCIENCES') THEN 'SCIENCE'
        WHEN UPPER(TRIM(b.category)) IN ('HISTORY', 'HISTORICAL') THEN 'HISTORY'
        WHEN TRIM(b.category) IS NULL THEN 'GENERAL'
```

Sample Assignment Report

```
        ELSE UPPER(TRIM(b.category))
    END as clean_category,
    CASE
        WHEN TRIM(UPPER(b.publisherName)) IS NULL
        THEN 'UNKNOWN PUBLISHER'
        ELSE TRIM(UPPER(b.publisherName))
    END as clean_publisherName,
    CASE
        WHEN TRIM(UPPER(b.publisherLocation)) IS NULL
        THEN 'UNKNOWN LOCATION'
        ELSE TRIM(UPPER(b.publisherLocation))
    END as clean_publisherLocation,
    CASE
        WHEN UPPER(TRIM(c.bookStatus)) IN ('AVAILABLE', 'AVAIL') THEN 'AVAILABLE'
        WHEN UPPER(TRIM(c.bookStatus)) IN ('CHECKED OUT', 'CHECKED_OUT', 'OUT')
THEN 'CHECKED_OUT'
        WHEN UPPER(TRIM(c.bookStatus)) IN ('RESERVED', 'RESERVE') THEN 'RESERVED'
        WHEN UPPER(TRIM(c.bookStatus)) IN ('DAMAGED', 'DAMAGE') THEN 'DAMAGED'
        WHEN UPPER(TRIM(c.bookStatus)) IN ('LOST', 'MISSING') THEN 'LOST'
        WHEN c.bookStatus IS NULL THEN 'AVAILABLE'
        ELSE UPPER(TRIM(c.bookStatus))
    END as clean_bookStatus,
    CASE
        WHEN UPPER(TRIM(c.bookCondition)) IN ('NEW', 'EXCELLENT') THEN
'EXCELLENT'
        WHEN UPPER(TRIM(c.bookCondition)) IN ('GOOD', 'FINE') THEN 'GOOD'
        WHEN UPPER(TRIM(c.bookCondition)) IN ('FAIR', 'AVERAGE') THEN 'FAIR'
        WHEN UPPER(TRIM(c.bookCondition)) IN ('POOR', 'BAD', 'DAMAGED') THEN
'POOR'
        WHEN c.bookCondition IS NULL THEN 'GOOD'
        ELSE UPPER(TRIM(c.bookCondition))
    END as clean_bookCondition,
    CASE WHEN b.bookTitle IS NULL OR LENGTH(TRIM(b.bookTitle)) < 2 THEN 'Y' ELSE
'N' END as title_cleaned,
    CASE WHEN b.bookImei IS NULL THEN 'Y' ELSE 'N' END as imei_generated,
    CASE WHEN b.publishYear IS NULL OR b.publishYear < 1800 OR b.publishYear >
EXTRACT(YEAR FROM SYSDATE) THEN 'Y' ELSE 'N' END as year_defaulted
FROM BOOK b
JOIN BOOK_COPY c ON b.bookID = c.bookID
WHERE b.bookID IS NOT NULL
  AND c.copyID IS NOT NULL;

-- ==================================================================
-- SECTION 2: CREATE SEQUENCE book_dim_seq
-- ==================================================================
CREATE SEQUENCE book_dim_seq
    START WITH 1000
    INCREMENT BY 1;

-- ==================================================================
-- SECTION 3: ETL (INITIAL LOADING)
-- ==================================================================
```

Sample Assignment Report

```
CREATE OR REPLACE PROCEDURE LOAD_BOOK_DIM_INITIAL AS
    v_count NUMBER;
    v_errors NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_count FROM BOOK_DIM;

    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('BOOK_DIM already contains data. Use
LOAD_BOOK_DIM_INCREMENTAL for updates.');
        RETURN;
    END IF;

    INSERT INTO BOOK_DIM (
        book_key, bookID, copyID, bookTitle, bookImei, language, author,
        publishYear, bookCategory, publisherName, publisherLocation,
        bookStatus, bookCondition
    )
    SELECT
        book_dim_seq.NEXTVAL,
        bookID,
        copyID,
        clean_bookTitle,
        clean_bookImei,
        clean_language,
        clean_author,
        clean_publishYear,
        clean_category,
        clean_publisherName,
        clean_publisherLocation,
        clean_bookStatus,
        clean_bookCondition
    FROM BOOK_STAGING_V;

    v_count := SQL%ROWCOUNT;

    SELECT COUNT(*) INTO v_errors
    FROM BOOK_STAGING_V
    WHERE title_cleaned = 'Y' OR imei_generated = 'Y' OR year_defaulted = 'Y';

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('BOOK_DIM initial load completed: ' || v_count || '
records inserted.');
    DBMS_OUTPUT.PUT_LINE('Data quality corrections applied: ' || v_errors || '
records had issues corrected.');
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in BOOK_DIM initial load: ' || SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

MEMBER_DIM

```
-- ====================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ====================================================================
CREATE OR REPLACE VIEW MEMBER_STAGING_V AS
SELECT
    m.memberID,
    CASE
        WHEN TRIM(UPPER(m.memberName)) IS NULL OR LENGTH(TRIM(m.memberName)) < 2
        THEN 'UNKNOWN MEMBER'
        ELSE TRIM(UPPER(m.memberName))
    END as clean_memberName,
    CASE
        WHEN m.memberEmail IS NULL OR NOT REGEXP_LIKE(m.memberEmail,
'^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$')
        THEN LOWER(REPLACE(m.memberID, ' ', '')) || '@library.temp'
        ELSE LOWER(TRIM(m.memberEmail))
    END as clean_memberEmail,
    NVL(m.dob, DATE '1900-01-01') as clean_memberDob,
    CASE
        WHEN m.contactNum IS NULL
        THEN '000-000-0000'
        ELSE REGEXP_REPLACE(m.contactNum, '[^0-9]', '')
    END as clean_contactNum,
    CASE
        WHEN UPPER(TRIM(m.memberCountry)) IN ('USA', 'UNITED STATES', 'US') THEN
'UNITED STATES'
        WHEN UPPER(TRIM(m.memberCountry)) IN ('UK', 'UNITED KINGDOM', 'BRITAIN')
THEN 'UNITED KINGDOM'
        WHEN UPPER(TRIM(m.memberCountry)) IN ('CANADA', 'CAN') THEN 'CANADA'
        WHEN TRIM(m.memberCountry) IS NULL THEN 'UNKNOWN'
        ELSE UPPER(TRIM(m.memberCountry))
    END as clean_memberCountry,
    CASE WHEN TRIM(UPPER(m.memberState)) IS NULL THEN 'UNKNOWN' ELSE
UPPER(TRIM(m.memberState)) END as clean_memberState,
    CASE WHEN TRIM(UPPER(m.memberCity)) IS NULL THEN 'UNKNOWN' ELSE
UPPER(TRIM(m.memberCity)) END as clean_memberCity,
    CASE WHEN m.memberZipcode IS NULL OR LENGTH(TRIM(m.memberZipcode)) < 5 THEN
'00000' ELSE TRIM(m.memberZipcode) END as clean_memberZipcode,
    CASE WHEN m.memberName IS NULL OR LENGTH(TRIM(m.memberName)) < 2 THEN 'Y'
ELSE 'N' END as name_cleaned,
    CASE WHEN m.memberEmail IS NULL OR NOT REGEXP_LIKE(m.memberEmail,
'^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$') THEN 'Y' ELSE 'N' END as
email_generated,
    CASE WHEN m.contactNum IS NULL THEN 'Y' ELSE 'N' END as phone_defaulted,
    CASE WHEN m.dob IS NULL THEN 'Y' ELSE 'N' END as dob_defaulted
FROM MEMBER m
WHERE m.memberID IS NOT NULL;

-- ====================================================================
-- SECTION 2: CREATE SEQUENCE member_dim_seq
```

Sample Assignment Report

```
-- ====================================================================
CREATE SEQUENCE member_dim_seq
    START WITH 1000
    INCREMENT BY 1;

-- ====================================================================
-- SECTION 3: ETL (INITIAL LOADING)
-- ====================================================================
CREATE OR REPLACE PROCEDURE LOAD_MEMBER_DIM_INITIAL AS
    v_count NUMBER;
    v_errors NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_count FROM MEMBER_DIM;
    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('MEMBER_DIM already contains data. Use
LOAD_MEMBER_DIM_INCREMENTAL for SCD2 updates.');
        RETURN;
    END IF;

    INSERT INTO MEMBER_DIM (
        member_key, memberID, memberName, memberEmail, dob, contactNum,
        memberCountry, memberState, memberCity, memberZipcode,
        effective_start_date, effective_end_date, is_current_flag
    )
    SELECT
        member_dim_seq.NEXTVAL,
        memberID, clean_memberName, clean_memberEmail, clean_memberDob,
clean_contactNum,
        clean_memberCountry, clean_memberState, clean_memberCity,
clean_memberZipcode,
        SYSDATE,
        DATE '9999-12-31',
        '1'
    FROM MEMBER_STAGING_V;

    v_count := SQL%ROWCOUNT;
    SELECT COUNT(*) INTO v_errors
    FROM MEMBER_STAGING_V
    WHERE name_cleaned = 'Y' OR email_generated = 'Y' OR phone_defaulted = 'Y' OR
dob_defaulted = 'Y';

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('MEMBER_DIM initial load completed: ' || v_count || '
records inserted.');
    DBMS_OUTPUT.PUT_LINE('Data quality corrections applied: ' || v_errors || '
records had issues corrected.');
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in MEMBER_DIM initial load: ' || SQLERRM);
        RAISE;
END;
```

Sample Assignment Report

```
/
```

DATE_DIM

```
-- ====================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ====================================================================
CREATE OR REPLACE VIEW DATE_STAGING_V AS
SELECT
    v_date,
    TO_CHAR(v_date, 'DD-MON-YYYY') as full_desc,
    TO_CHAR(v_date, 'DY') as day_week,
    TO_NUMBER(TO_CHAR(v_date,'DD')) as day_num_month,
    TO_NUMBER(TO_CHAR(v_date,'DDD')) as day_num_year,
    CASE WHEN v_date = LAST_DAY(v_date) THEN 'Y' ELSE 'N' END as last_day_ind,
    NEXT_DAY(v_date, 'SATURDAY') as cal_week_end_date,
    TO_NUMBER(TO_CHAR(v_date,'WW')) as cal_week_year,
    TO_CHAR(v_date,'Month') as cal_month_name,
    TO_NUMBER(TO_CHAR(v_date,'MM')) as cal_month_year,
    TO_CHAR(v_date,'YYYY-MM') as cal_year_month,
    TO_CHAR(v_date,'Q') as cal_quarter,
    TO_CHAR(v_date,'YYYY') || '-Q' || TO_CHAR(v_date,'Q') as cal_year_quarter,
    TO_NUMBER(TO_CHAR(v_date,'YYYY')) as cal_year,
    'N' as holiday_ind,
    CASE WHEN TO_CHAR(v_date,'DY','NLS_DATE_LANGUAGE=ENGLISH') IN ('SAT','SUN')
THEN 'N' ELSE 'Y' END as weekday_ind,
    'N' as exam_week_ind,
    'N' as study_week_ind,
    NULL as festive_event
FROM (
    SELECT TO_DATE('2016-01-01','YYYY-MM-DD') + LEVEL - 1 AS v_date
    FROM dual
    CONNECT BY LEVEL <= 5475 -- Generate ~15 years of dates
)
WHERE v_date IS NOT NULL;

-- ====================================================================
-- SECTION 2: CREATE SEQUENCE DATE_DIM_SEQ
-- ====================================================================
CREATE SEQUENCE DATE_DIM_SEQ
    START WITH 1
    INCREMENT BY 1;

-- ====================================================================
-- SECTION 3: ETL (INITIAL LOADING)
-- ====================================================================
CREATE OR REPLACE PROCEDURE LOAD_DATE_DIM_INITIAL AS
    v_inserted_count NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_inserted_count FROM DATE_DIM;
```

Sample Assignment Report

```
    IF v_inserted_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('DATE_DIM already contains data. Use
LOAD_DATE_DIM_INCREMENTAL for updates.');
        RETURN;
    END IF;

    INSERT INTO DATE_DIM (
        DATE_KEY, CAL_DATE, FULL_DESC, DAY_WEEK, DAY_NUM_MONTH, DAY_NUM_YEAR,
        LAST_DAY_IND, CAL_WEEK_END_DATE, CAL_WEEK_YEAR, CAL_MONTH_NAME,
        CAL_MONTH_YEAR, CAL_YEAR_MONTH, CAL_QUARTER, CAL_YEAR_QUARTER, CAL_YEAR,
        HOLIDAY_IND, WEEKDAY_IND, EXAM_WEEK_IND, STUDY_WEEK_IND, FESTIVE_EVENT
    )
    SELECT
        DATE_DIM_SEQ.NEXTVAL,
        v_date, full_desc, day_week, day_num_month, day_num_year,
        last_day_ind, cal_week_end_date, cal_week_year, cal_month_name,
        cal_month_year, cal_year_month, cal_quarter, cal_year_quarter, cal_year,
        holiday_ind, weekday_ind, exam_week_ind, study_week_ind, festive_event
    FROM DATE_STAGING_V
    WHERE v_date < TO_DATE('2026-01-01', 'YYYY-MM-DD');

    v_inserted_count := SQL%ROWCOUNT;

    UPDATE DATE_DIM
    SET STUDY_WEEK_IND = 'Y'
    WHERE (TO_CHAR(CAL_DATE, 'MM-DD') >= '01-06' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '01-12')
       OR (TO_CHAR(CAL_DATE, 'MM-DD') >= '05-19' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '05-25')
       OR (TO_CHAR(CAL_DATE, 'MM-DD') >= '09-29' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '10-05');

    UPDATE DATE_DIM
    SET EXAM_WEEK_IND = 'Y'
    WHERE (TO_CHAR(CAL_DATE, 'MM-DD') >= '01-13' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '01-19')
       OR (TO_CHAR(CAL_DATE, 'MM-DD') >= '05-26' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '06-08')
       OR (TO_CHAR(CAL_DATE, 'MM-DD') >= '10-06' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '10-19');

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('DATE_DIM initial load completed: ' || v_inserted_count
|| ' records inserted.');
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in DATE_DIM initial load: ' || SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

STAFF_DIM

```
-- ====================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ====================================================================
CREATE OR REPLACE VIEW STAFF_STAGING_V AS
SELECT
    s.staffID,
    CASE
        WHEN TRIM(UPPER(s.staffName)) IS NULL OR LENGTH(TRIM(s.staffName)) < 2
        THEN 'UNKNOWN STAFF'
        ELSE TRIM(UPPER(s.staffName))
    END as clean_staffName,
    CASE
        WHEN s.staffEmail IS NULL OR NOT REGEXP_LIKE(s.staffEmail,
'^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$')
        THEN LOWER(REPLACE(s.staffID, ' ', '')) || '@library.org'
        ELSE LOWER(TRIM(s.staffEmail))
    END as clean_staffEmail,
    CASE
        WHEN s.staffPhoneNum IS NULL
        THEN '000-000-0000'
        ELSE REGEXP_REPLACE(s.staffPhoneNum, '[^0-9]', '')
        END as clean_staffPhoneNum,
    CASE
        WHEN UPPER(TRIM(s.department)) IN ('ADMIN', 'ADMINISTRATION',
'ADMINISTRATIVE')
        THEN 'ADMINISTRATION'
        WHEN UPPER(TRIM(s.department)) IN ('CIRCULATION', 'CIRC', 'CHECKOUT')
        THEN 'CIRCULATION'
        WHEN UPPER(TRIM(s.department)) IN ('REFERENCE', 'REF', 'RESEARCH')
        THEN 'REFERENCE'
        WHEN UPPER(TRIM(s.department)) IN ('TECHNICAL', 'TECH', 'IT', 'SYSTEMS')
        THEN 'TECHNICAL SERVICES'
        WHEN UPPER(TRIM(s.department)) IN ('ACQUISITIONS', 'ACQU', 'PURCHASING')
        THEN 'ACQUISITIONS'
        WHEN UPPER(TRIM(s.department)) IN ('CATALOGING', 'CATALOG', 'CAT')
        THEN 'CATALOGING'
        WHEN s.department IS NULL
        THEN 'GENERAL'
        ELSE UPPER(TRIM(s.department))
    END as clean_department,
    CASE
        WHEN UPPER(TRIM(s.position)) IN ('LIBRARIAN', 'HEAD LIBRARIAN', 'CHIEF
LIBRARIAN')
        THEN 'LIBRARIAN'
        WHEN UPPER(TRIM(s.position)) IN ('ASSISTANT', 'LIBRARY ASSISTANT',
'ASST')
        THEN 'LIBRARY ASSISTANT'
        WHEN UPPER(TRIM(s.position)) IN ('MANAGER', 'SUPERVISOR', 'HEAD')
        THEN 'MANAGER'
        WHEN UPPER(TRIM(s.position)) IN ('CLERK', 'OFFICE CLERK', 'DATA ENTRY')
```

Sample Assignment Report

```
        THEN 'CLERK'
        WHEN UPPER(TRIM(s.position)) IN ('TECHNICIAN', 'TECH', 'IT SUPPORT')
        THEN 'TECHNICIAN'
        WHEN UPPER(TRIM(s.position)) IN ('INTERN', 'STUDENT', 'TEMP')
        THEN 'TEMPORARY'
        WHEN s.position IS NULL
        THEN 'STAFF'
        ELSE UPPER(TRIM(s.position))
    END as clean_position,
    CASE
        WHEN s.hireDate IS NULL OR s.hireDate > SYSDATE
        THEN SYSDATE WHEN s.hireDate < DATE '1950-01-01'
        THEN DATE '1950-01-01'
        ELSE s.hireDate
    END as clean_hireDate,
    CASE
        WHEN s.staffName IS NULL OR LENGTH(TRIM(s.staffName)) < 2
        THEN 'Y' ELSE 'N'
    END as name_cleaned,
    CASE
        WHEN s.staffEmail IS NULL OR NOT REGEXP_LIKE(s.staffEmail,
'^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$')
        THEN 'Y'
        ELSE 'N'
    END as email_generated,
    CASE
        WHEN s.staffPhoneNum IS NULL
        THEN 'Y'
        ELSE 'N'
    END as phone_defaulted,
    CASE
        WHEN s.hireDate IS NULL OR s.hireDate > SYSDATE OR s.hireDate < DATE
'1950-01-01'
        THEN 'Y'
        ELSE 'N'
    END as date_corrected
FROM STAFF s
WHERE s.staffID IS NOT NULL;

-- ====================================================================
-- SECTION 2: CREATE SEQUENCE STAFF_DIM_SEQ
-- ====================================================================
CREATE SEQUENCE staff_dim_seq
    START WITH 1000
    INCREMENT BY 1;

-- ====================================================================
-- SECTION 3: ETL (INITIAL LOADING)
-- ====================================================================
CREATE OR REPLACE PROCEDURE LOAD_STAFF_DIM_INITIAL AS
    v_count NUMBER;
    v_errors NUMBER := 0;
```

Sample Assignment Report

```
BEGIN
    SELECT COUNT(*) INTO v_count FROM STAFF_DIM;
    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('STAFF_DIM already contains data. Use
LOAD_STAFF_DIM_INCREMENTAL for SCD2 updates.');
        RETURN;
    END IF;

    INSERT INTO STAFF_DIM (
        staff_key, staffID, staffName, staffEmail, staffPhoneNum,
        department, position, hireDate,
        effective_start_date, effective_end_date, is_current_flag
    )
    SELECT
        staff_dim_seq.NEXTVAL,
        staffID, clean_staffName, clean_staffEmail, clean_staffPhoneNum,
        clean_department, clean_position, clean_hireDate,
        SYSDATE,
        DATE '9999-12-31',
        '1'
    FROM STAFF_STAGING_V;

    v_count := SQL%ROWCOUNT;
    SELECT COUNT(*) INTO v_errors
    FROM STAFF_STAGING_V
    WHERE name_cleaned = 'Y' OR email_generated = 'Y' OR phone_defaulted = 'Y' OR
date_corrected = 'Y';

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('STAFF_DIM initial load completed: ' || v_count || '
records inserted.');
    DBMS_OUTPUT.PUT_LINE('Data quality corrections applied: ' || v_errors || '
records had issues corrected');
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in STAFF_DIM initial load: ' || SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

SUPPLIER_DIM

```
-- ==================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ==================================================================
CREATE OR REPLACE VIEW SUPPLIER_STAGING_V AS
SELECT
    s.supplierID,
    CASE
        WHEN TRIM(UPPER(s.supplierName)) IS NULL OR LENGTH(TRIM(s.supplierName))
```

```
< 2
        THEN 'UNKNOWN SUPPLIER'
        ELSE TRIM(UPPER(s.supplierName))
        END as clean_supplierName,
    CASE
        WHEN UPPER(TRIM(s.supplierLocation)) IN ('NYC', 'NEW YORK CITY')
        THEN 'NEW YORK, NY'
        WHEN UPPER(TRIM(s.supplierLocation)) IN ('LA', 'LOS ANGELES')
        THEN 'LOS ANGELES, CA'
        WHEN UPPER(TRIM(s.supplierLocation)) IN ('CHI', 'CHICAGO')
        THEN 'CHICAGO, IL'
        WHEN UPPER(TRIM(s.supplierLocation)) IN ('SF', 'SAN FRANCISCO')
        THEN 'SAN FRANCISCO, CA'
        WHEN UPPER(TRIM(s.supplierLocation)) LIKE '%UK%' OR
UPPER(TRIM(s.supplierLocation)) LIKE '%UNITED KINGDOM%'
        THEN REGEXP_REPLACE(UPPER(TRIM(s.supplierLocation)), '(UK|UNITED
KINGDOM)', 'UNITED KINGDOM')
        WHEN s.supplierLocation IS NULL
        THEN 'UNKNOWN LOCATION'
        ELSE UPPER(TRIM(s.supplierLocation))
    END as clean_supplierLocation,
    CASE
        WHEN s.supplierContactNum IS NULL
        THEN '000-000-0000'
        WHEN REGEXP_LIKE(s.supplierContactNum, '^\+')
        THEN s.supplierContactNum
        ELSE REGEXP_REPLACE(s.supplierContactNum, '[^0-9]', '')
    END as clean_supplierContactNum,
    CASE
        WHEN s.supplierEmail IS NULL OR NOT REGEXP_LIKE(s.supplierEmail,
'^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$')
        THEN 'contact@' || LOWER(REPLACE(REPLACE(s.supplierName, ' ', ''), '''',
'')) || '.com'
        ELSE LOWER(TRIM(s.supplierEmail))
    END as clean_supplierEmail,
    CASE
        WHEN s.supplierName IS NULL OR LENGTH(TRIM(s.supplierName)) < 2
        THEN 'Y'
        ELSE 'N'
    END as name_cleaned,
    CASE
        WHEN s.supplierLocation IS NULL
        THEN 'Y'
        ELSE 'N'
    END as location_defaulted,
    CASE
        WHEN s.supplierContactNum IS NULL
        THEN 'Y'
        ELSE 'N'
    END as phone_defaulted,
    CASE
        WHEN s.supplierEmail IS NULL OR NOT REGEXP_LIKE(s.supplierEmail,
```

Sample Assignment Report

```
'^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$')
        THEN 'Y'
        ELSE 'N'
    END as email_generated
FROM SUPPLIER s
WHERE s.supplierID IS NOT NULL;

-- ==================================================================
-- SECTION 2: CREATE SEQUENCE supplier_dim_seq
-- ==================================================================
CREATE SEQUENCE supplier_dim_seq
    START WITH 1000
    INCREMENT BY 1;

-- ==================================================================
-- SECTION 3: ETL (INITIAL LOADING)
-- ==================================================================
CREATE OR REPLACE PROCEDURE LOAD_SUPPLIER_DIM_INITIAL AS
    v_count NUMBER;
    v_errors NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_count FROM SUPPLIER_DIM;
    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('SUPPLIER_DIM already contains data. Use
LOAD_SUPPLIER_DIM_INCREMENTAL for SCD2 updates.');
        RETURN;
    END IF;

    INSERT INTO SUPPLIER_DIM (
        supplier_key, supplierID, supplierName, supplierLocation,
        supplierContactNum, supplierEmail,
        effective_start_date, effective_end_date, is_current_flag
    )
    SELECT
        supplier_dim_seq.NEXTVAL,
        supplierID, clean_supplierName, clean_supplierLocation,
        clean_supplierContactNum, clean_supplierEmail,
        SYSDATE,
        DATE '9999-12-31',
        '1'
    FROM SUPPLIER_STAGING_V;

    v_count := SQL%ROWCOUNT;
    SELECT COUNT(*) INTO v_errors
    FROM SUPPLIER_STAGING_V
    WHERE name_cleaned = 'Y' OR location_defaulted = 'Y' OR phone_defaulted = 'Y'
OR email_generated = 'Y';

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('SUPPLIER_DIM initial load completed: ' || v_count || '
records inserted.');
    DBMS_OUTPUT.PUT_LINE('Data quality corrections applied: ' || v_errors || '
```

Sample Assignment Report

```
records had issues corrected.');
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in SUPPLIER_DIM initial load: ' || SQLERRM);
        RAISE;
END;
/
```

RESERVATION_FACT

```
-- ====================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ====================================================================
CREATE OR REPLACE VIEW RESERVATION_STAGING_V AS
SELECT
    r.reserveID,
    r.reserveStartDate,
    r.reserveEndDate,
    r.memberID,
    r.staffID,
    rd.copyID,
    CASE
        WHEN UPPER(TRIM(rd.reserveStatus)) IN ('ACTIVE', 'PENDING', 'WAITING')
THEN 'ACTIVE'
        WHEN UPPER(TRIM(rd.reserveStatus)) IN ('FULFILLED', 'COMPLETED',
'PICKED_UP') THEN 'FULFILLED'
        WHEN UPPER(TRIM(rd.reserveStatus)) IN ('EXPIRED', 'TIMEOUT') THEN
'EXPIRED'
        WHEN UPPER(TRIM(rd.reserveStatus)) IN ('CANCELLED', 'CANCELED') THEN
'CANCELLED'
        WHEN rd.reserveStatus IS NULL THEN 'ACTIVE' -- Default null status to
ACTIVE
        ELSE UPPER(TRIM(rd.reserveStatus))
    END as clean_reserveStatus,
    CASE
        WHEN r.reserveEndDate IS NOT NULL AND r.reserveStartDate IS NOT NULL THEN
            CASE
                WHEN (r.reserveEndDate - r.reserveStartDate) < 0 THEN 7
                WHEN (r.reserveEndDate - r.reserveStartDate) > 30 THEN 30
                ELSE (r.reserveEndDate - r.reserveStartDate)
            END
        WHEN r.reserveEndDate IS NULL AND r.reserveStartDate IS NOT NULL THEN
            7
        ELSE NULL
    END as clean_reservationDuration,
    CASE
        WHEN r.reserveEndDate IS NULL AND r.reserveStartDate IS NOT NULL
        THEN r.reserveStartDate + 7
        ELSE r.reserveEndDate
    END as clean_reserveEndDate,
```

Sample Assignment Report

```
    CASE WHEN r.reserveEndDate IS NULL THEN 'Y' ELSE 'N' END as
enddate_generated,
    CASE WHEN rd.reserveStatus IS NULL THEN 'Y' ELSE 'N' END as status_defaulted,
    CASE WHEN r.reserveEndDate IS NOT NULL AND (r.reserveEndDate -
r.reserveStartDate) < 0 THEN 'Y' ELSE 'N' END as duration_fixed
FROM RESERVATION r
JOIN RESERVATION_DETAILS rd ON r.reserveID = rd.reserveID
WHERE r.reserveID IS NOT NULL
  AND r.reserveStartDate IS NOT NULL
  AND r.memberID IS NOT NULL
  AND r.staffID IS NOT NULL
  AND rd.copyID IS NOT NULL;


-- ==================================================================
-- SECTION 2: ETL (INITIAL LOADING)
-- ==================================================================
CREATE OR REPLACE PROCEDURE LOAD_RESERVATION_FACT_INITIAL AS
    v_count NUMBER;
    v_errors NUMBER := 0;
    v_orphaned NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_count FROM RESERVATION_FACT;

    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('RESERVATION_FACT already contains data. Use
LOAD_RESERVATION_FACT_INC for updates.');
        RETURN;
    END IF;

    INSERT INTO RESERVATION_FACT (
        member_key, book_key, staff_key, reserve_start_date_key,
        reserve_end_date_key, reserveID, reservationStatus, reservationDuration
    )
    SELECT
        M.member_key,
        B.book_key,
        S.staff_key,
        D_START.date_key AS reserve_start_date_key,
        D_END.date_key AS reserve_end_date_key,
        rs.reserveID,
        rs.clean_reserveStatus,
        rs.clean_reservationDuration
    FROM RESERVATION_STAGING_V rs
    JOIN DATE_DIM D_START ON rs.reserveStartDate = D_START.cal_date
    JOIN DATE_DIM D_END ON rs.clean_reserveEndDate = D_END.cal_date
    JOIN MEMBER_DIM M ON rs.memberID = M.memberID AND M.is_current_flag = '1'
    JOIN BOOK_DIM B ON rs.copyID = B.copyID
    JOIN STAFF_DIM S ON rs.staffID = S.staffID AND S.is_current_flag = '1';

    v_count := SQL%ROWCOUNT;
```

Sample Assignment Report

```
    SELECT COUNT(*) INTO v_errors
    FROM RESERVATION_STAGING_V
    WHERE enddate_generated = 'Y' OR status_defaulted = 'Y' OR duration_fixed =
'Y';

    SELECT COUNT(*) INTO v_orphaned
    FROM RESERVATION_STAGING_V rs
    WHERE NOT EXISTS (SELECT 1 FROM DATE_DIM D WHERE rs.reserveStartDate =
D.cal_date)
       OR NOT EXISTS (SELECT 1 FROM DATE_DIM D WHERE rs.clean_reserveEndDate =
D.cal_date)
       OR NOT EXISTS (SELECT 1 FROM MEMBER_DIM M WHERE rs.memberID = M.memberID
AND M.is_current_flag = '1')
       OR NOT EXISTS (SELECT 1 FROM BOOK_DIM B WHERE rs.copyID = B.copyID)
       OR NOT EXISTS (SELECT 1 FROM STAFF_DIM S WHERE rs.staffID = S.staffID AND
S.is_current_flag = '1');

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('RESERVATION_FACT initial load completed: ' || v_count
|| ' records inserted.');
    DBMS_OUTPUT.PUT_LINE('Data quality corrections: ' || v_errors || ' records
corrected.');
    IF v_orphaned > 0 THEN
        DBMS_OUTPUT.PUT_LINE('WARNING: ' || v_orphaned || ' records skipped due
to missing dimension references.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in RESERVATION_FACT initial load: ' ||
SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

LOAN_FACT

```
-- ====================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ====================================================================
-- VIEW 1: LOAN_STAGING_V
-- This view joins raw loan tables and performs data scrubbing.
CREATE OR REPLACE VIEW LOAN_STAGING_V AS
SELECT
    l.loanID,
    l.loanDate,
    l.memberID,
    l.staffID,
    ld.copyID,
    CASE
        WHEN ld.loanDuration IS NULL OR ld.loanDuration <= 0
```

```
        THEN 14
        WHEN ld.loanDuration > 365
        THEN 365
        ELSE ld.loanDuration
    END as clean_loanDuration,
    CASE
        WHEN ld.returnDate IS NOT NULL AND ld.dueDate IS NOT NULL THEN
            CASE
                WHEN ld.returnDate > ld.dueDate THEN (ld.returnDate - ld.dueDate)
                ELSE 0
            END
        ELSE NULL
    END as clean_overdueDays,
    CASE
        WHEN UPPER(TRIM(ld.loanStatus)) IN ('ACTIVE', 'CURRENT', 'OUT') THEN
'ACTIVE'
        WHEN UPPER(TRIM(ld.loanStatus)) IN ('RETURNED', 'COMPLETE', 'COMPLETED')
THEN 'RETURNED'
        WHEN UPPER(TRIM(ld.loanStatus)) IN ('OVERDUE', 'LATE') THEN 'OVERDUE'
        WHEN UPPER(TRIM(ld.loanStatus)) IN ('LOST', 'MISSING') THEN 'LOST'
        WHEN ld.loanStatus IS NULL THEN 'ACTIVE'
        ELSE UPPER(TRIM(ld.loanStatus))
    END as clean_loanStatus,
    l.loanID as fine_lookup_id,
    CASE WHEN ld.loanDuration IS NULL OR ld.loanDuration <= 0 OR ld.loanDuration
> 365 THEN 'Y' ELSE 'N' END as duration_corrected,
    CASE WHEN ld.loanStatus IS NULL THEN 'Y' ELSE 'N' END as status_defaulted
FROM LOAN l
JOIN LOAN_DETAILS ld ON l.loanID = ld.loanID
WHERE l.loanID IS NOT NULL
  AND l.loanDate IS NOT NULL
  AND l.memberID IS NOT NULL
  AND l.staffID IS NOT NULL
  AND ld.copyID IS NOT NULL;

-- VIEW 2: FINE_SUMMARY_V
-- This view aggregates fine data for each loan.
CREATE OR REPLACE VIEW FINE_SUMMARY_V AS
SELECT
    loanID,
    SUM(CASE WHEN fineAmt IS NOT NULL AND fineAmt >= 0 THEN fineAmt ELSE 0 END)
as totalFineAmt,
    CASE
        WHEN COUNT(fineID) > 0 AND COUNT(CASE WHEN UPPER(TRIM(fineStatus)) IN
('COMPLETED', 'PAID') THEN 1 END) = COUNT(fineID) THEN 1
        WHEN COUNT(fineID) > 0 THEN 0
        ELSE NULL
    END as paid_flag
FROM Fine
WHERE loanID IS NOT NULL
GROUP BY loanID;
```

Sample Assignment Report

```
-- ====================================================================
-- SECTION 2: ETL (INITIAL LOADING)
-- ====================================================================
CREATE OR REPLACE PROCEDURE LOAD_LOAN_FACT_INITIAL AS
    v_count NUMBER;
    v_errors NUMBER := 0;
    v_orphaned NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_count FROM LOAN_FACT;

    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('LOAN_FACT already contains data. Use
LOAD_LOAN_FACT_INCREMENTAL for updates.');
        RETURN;
    END IF;

    INSERT INTO LOAN_FACT (
        member_key, book_key, staff_key, date_key, loanID,
        loanDuration, overdueDays, loanStatus, totalFine, fine_paid_flag
    )
    SELECT
        M.member_key,
        B.book_key,
        S.staff_key,
        D.date_key,
        ls.loanID,
        ls.clean_loanDuration,
        ls.clean_overdueDays,
        ls.clean_loanStatus,
        COALESCE(f.totalFineAmt, 0) AS totalFine,
        f.paid_flag AS fine_paid_flag
    FROM LOAN_STAGING_V ls
    JOIN DATE_DIM D ON ls.loanDate = D.cal_date
    JOIN MEMBER_DIM M ON ls.memberID = M.memberID AND M.is_current_flag = '1'
    JOIN BOOK_DIM B ON ls.copyID = B.copyID
    JOIN STAFF_DIM S ON ls.staffID = S.staffID AND S.is_current_flag = '1'
    LEFT JOIN FINE_SUMMARY_V f ON ls.fine_lookup_id = f.loanID;

    v_count := SQL%ROWCOUNT;

    SELECT COUNT(*) INTO v_errors
    FROM LOAN_STAGING_V
    WHERE duration_corrected = 'Y' OR status_defaulted = 'Y';

    SELECT COUNT(*) INTO v_orphaned
    FROM LOAN_STAGING_V ls
    WHERE NOT EXISTS (SELECT 1 FROM DATE_DIM D WHERE ls.loanDate = D.cal_date)
       OR NOT EXISTS (SELECT 1 FROM MEMBER_DIM M WHERE ls.memberID = M.memberID
AND M.is_current_flag = '1')
       OR NOT EXISTS (SELECT 1 FROM BOOK_DIM B WHERE ls.copyID = B.copyID)
       OR NOT EXISTS (SELECT 1 FROM STAFF_DIM S WHERE ls.staffID = S.staffID AND
```

Sample Assignment Report

```
S.is_current_flag = '1');

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('LOAN_FACT initial load completed: ' || v_count || '
records inserted.');
    DBMS_OUTPUT.PUT_LINE('Data quality corrections: ' || v_errors || ' records
corrected.');
    IF v_orphaned > 0 THEN
        DBMS_OUTPUT.PUT_LINE('WARNING: ' || v_orphaned || ' records skipped due
to missing dimension references.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in LOAN_FACT initial load: ' || SQLERRM);
        RAISE;
END;
/
```

PURCHASE_FACT

Sample Assignment Report

```
-- ==================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ==================================================================
CREATE OR REPLACE VIEW PURCHASE_STAGING_V AS
SELECT
    p.purchaseID,
    p.purchaseDate,
    p.bookID,
    p.staffID,
    p.supplierID,
    CASE
        WHEN p.purchaseQty IS NULL OR p.purchaseQty <= 0
        THEN 1
        WHEN p.purchaseQty > 1000
        THEN 1000
        ELSE p.purchaseQty
    END as clean_purchaseQty,
    CASE
        WHEN p.purchaseUnitCost IS NULL OR p.purchaseUnitCost <= 0
        THEN 25.00
        WHEN p.purchaseUnitCost > 500
        THEN 500.00
        ELSE p.purchaseUnitCost
    END as clean_purchaseUnitCost,
    CASE WHEN p.purchaseQty IS NULL OR p.purchaseQty <= 0 OR p.purchaseQty > 1000
THEN 'Y' ELSE 'N' END as qty_corrected,
    CASE WHEN p.purchaseUnitCost IS NULL OR p.purchaseUnitCost <= 0 OR
p.purchaseUnitCost > 500 THEN 'Y' ELSE 'N' END as cost_corrected
FROM PURCHASE p
WHERE p.purchaseID IS NOT NULL
```

```
  AND p.purchaseDate IS NOT NULL
  AND p.bookID IS NOT NULL
  AND p.staffID IS NOT NULL
  AND p.supplierID IS NOT NULL;

-- ====================================================================
-- SECTION 2: ETL (INITIAL LOADING)
-- ====================================================================
CREATE OR REPLACE PROCEDURE LOAD_PURCHASE_FACT_INITIAL AS
    v_count NUMBER;
    v_errors NUMBER := 0;
    v_orphaned NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_count FROM PURCHASE_FACT;

    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('PURCHASE_FACT already contains data. Use
LOAD_PURCHASE_FACT_INCREMENTAL for updates.');
        RETURN;
    END IF;

    -- FIX: A WITH clause is used to de-duplicate the BOOK_DIM on-the-fly to
resolve the grain mismatch.
    INSERT INTO PURCHASE_FACT (
        date_key, book_key, staff_key, supplier_key, purchaseID,
        purchaseQuantity, purchaseUnitCost, purchaseTotalCost
    )
    WITH BOOK_DIM_DEDUPED AS (
        SELECT
            bookID,
            MIN(book_key) as book_key -- Arbitrarily pick the first book_key for
each bookID
        FROM BOOK_DIM
        GROUP BY bookID
    )
    SELECT
        D.date_key,
        B.book_key,
        S.staff_key,
        SUP.supplier_key,
        ps.purchaseID,
        ps.clean_purchaseQty,
        ps.clean_purchaseUnitCost,
        (ps.clean_purchaseQty * ps.clean_purchaseUnitCost) AS purchaseTotalCost
    FROM PURCHASE_STAGING_V ps
    JOIN DATE_DIM D ON ps.purchaseDate = D.cal_date
    JOIN BOOK_DIM_DEDUPED B ON ps.bookID = B.bookID -- Corrected join
    JOIN STAFF_DIM S ON ps.staffID = S.staffID AND S.is_current_flag = '1'
    JOIN SUPPLIER_DIM SUP ON ps.supplierID = SUP.supplierID AND
SUP.is_current_flag = '1';

    v_count := SQL%ROWCOUNT;
```

Sample Assignment Report

```
    SELECT COUNT(*) INTO v_errors
    FROM PURCHASE_STAGING_V
    WHERE qty_corrected = 'Y' OR cost_corrected = 'Y';

    SELECT COUNT(*) INTO v_orphaned
    FROM PURCHASE_STAGING_V ps
    WHERE NOT EXISTS (SELECT 1 FROM DATE_DIM D WHERE ps.purchaseDate =
D.cal_date)
       OR NOT EXISTS (SELECT 1 FROM BOOK_DIM B WHERE ps.bookID = B.bookID)
       OR NOT EXISTS (SELECT 1 FROM STAFF_DIM S WHERE ps.staffID = S.staffID AND
S.is_current_flag = '1')
       OR NOT EXISTS (SELECT 1 FROM SUPPLIER_DIM SUP WHERE ps.supplierID =
SUP.supplierID AND SUP.is_current_flag = '1');

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('PURCHASE_FACT initial load completed: ' || v_count || '
records inserted.');
    DBMS_OUTPUT.PUT_LINE('Data quality corrections: ' || v_errors || ' records
corrected.');
    IF v_orphaned > 0 THEN
        DBMS_OUTPUT.PUT_LINE('WARNING: ' || v_orphaned || ' records skipped due
to missing dimension references.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in PURCHASE_FACT initial load: ' || SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

SALES_FACT

```
-- ==================================================================
-- SECTION 1: CORE ETL TRANSFORMATION LOGIC (VIEWS)
-- ==================================================================
CREATE OR REPLACE VIEW SALES_STAGING_V AS
SELECT
    o.orderID,
    o.orderDate,
    o.memberID,
    o.staffID,
    od.bookID,
    CASE
        WHEN od.orderQty IS NULL OR od.orderQty <= 0
        THEN 1  -- Default to 1 for invalid quantities
        WHEN od.orderQty > 50
        THEN 50
        ELSE od.orderQty
    END as clean_orderQty,
```

```
    CASE
        WHEN od.orderUnitPrice IS NULL OR od.orderUnitPrice <= 0
        THEN 20.00 -- Default reasonable unit price
        WHEN od.orderUnitPrice > 200
        THEN 200.00
        ELSE od.orderUnitPrice
    END as clean_orderUnitPrice,
    CASE WHEN od.orderQty IS NULL OR od.orderQty <= 0 OR od.orderQty > 50 THEN
'Y' ELSE 'N' END as qty_corrected,
    CASE WHEN od.orderUnitPrice IS NULL OR od.orderUnitPrice <= 0 OR
od.orderUnitPrice > 200 THEN 'Y' ELSE 'N' END as price_corrected
FROM ORDERS o
JOIN ORDER_DETAILS od ON o.orderID = od.orderID
WHERE o.orderID IS NOT NULL
  AND o.orderDate IS NOT NULL
  AND o.memberID IS NOT NULL
  AND o.staffID IS NOT NULL
  AND od.bookID IS NOT NULL;


-- ==================================================================
-- SECTION 2: ETL (INITIAL LOADING)
-- ==================================================================
CREATE OR REPLACE PROCEDURE LOAD_SALES_FACT_INITIAL AS
    v_count NUMBER;
    v_errors NUMBER := 0;
    v_orphaned NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_count FROM SALES_FACT;

    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('SALES_FACT already contains data. Use
LOAD_SALES_FACT_INCREMENTAL for updates.');
        RETURN;
    END IF;

    -- Insert with comprehensive validation
    -- FIX: A WITH clause is used to de-duplicate the BOOK_DIM on-the-fly.
    -- This resolves the "grain mismatch" and prevents a single sale from being
multiplied
    -- by the number of physical book copies.
    INSERT INTO SALES_FACT (
        date_key, member_key, book_key, staff_key, orderID,
        orderQty, orderUnitPrice, orderTotalPrice
    )
    WITH BOOK_DIM_DEDUPED AS (
        SELECT
            bookID,
            MIN(book_key) as book_key -- Arbitrarily pick the first book_key for
each bookID
        FROM BOOK_DIM
        GROUP BY bookID
```

Sample Assignment Report

```
    )
    SELECT
        D.date_key,
        M.member_key,
        B.book_key,
        S.staff_key,
        ss.orderID,
        ss.clean_orderQty,
        ss.clean_orderUnitPrice,
        (ss.clean_orderQty * ss.clean_orderUnitPrice) AS orderTotalPrice
    FROM SALES_STAGING_V ss
    JOIN DATE_DIM D ON ss.orderDate = D.cal_date
    JOIN MEMBER_DIM M ON ss.memberID = M.memberID AND M.is_current_flag = '1'
    JOIN BOOK_DIM_DEDUPED B ON ss.bookID = B.bookID -- Corrected join
    JOIN STAFF_DIM S ON ss.staffID = S.staffID AND S.is_current_flag = '1';

    v_count := SQL%ROWCOUNT;

    SELECT COUNT(*) INTO v_errors
    FROM SALES_STAGING_V
    WHERE qty_corrected = 'Y' OR price_corrected = 'Y';

    SELECT COUNT(*) INTO v_orphaned
    FROM SALES_STAGING_V ss
    WHERE NOT EXISTS (SELECT 1 FROM DATE_DIM D WHERE ss.orderDate = D.cal_date)
       OR NOT EXISTS (SELECT 1 FROM MEMBER_DIM M WHERE ss.memberID = M.memberID
AND M.is_current_flag = '1')
       OR NOT EXISTS (SELECT 1 FROM BOOK_DIM B WHERE ss.bookID = B.bookID)
       OR NOT EXISTS (SELECT 1 FROM STAFF_DIM S WHERE ss.staffID = S.staffID AND
S.is_current_flag = '1');

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('SALES_FACT initial load completed: ' || v_count || '
records inserted.');
    DBMS_OUTPUT.PUT_LINE('Data quality corrections: ' || v_errors || ' records
corrected.');
    IF v_orphaned > 0 THEN
        DBMS_OUTPUT.PUT_LINE('WARNING: ' || v_orphaned || ' records skipped due
to missing dimension references.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in SALES_FACT initial load: ' || SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

2.2 Script for subsequent loading

BOOK_DIM

```
CREATE OR REPLACE PROCEDURE LOAD_BOOK_DIM_INCREMENTAL AS
    v_count NUMBER := 0;
    v_updated NUMBER := 0;
    v_errors NUMBER := 0;
BEGIN
    INSERT INTO BOOK_DIM (
        book_key, bookID, copyID, bookTitle, bookImei, language, author,
        publishYear, bookCategory, publisherName, publisherLocation,
        bookStatus, bookCondition
    )
    SELECT
        book_dim_seq.NEXTVAL,
        s.bookID, s.copyID, s.clean_bookTitle, s.clean_bookImei,
        s.clean_language, s.clean_author, s.clean_publishYear, s.clean_category,
        s.clean_publisherName, s.clean_publisherLocation, s.clean_bookStatus,
        s.clean_bookCondition
    FROM BOOK_STAGING_V s
    WHERE NOT EXISTS (
        SELECT 1 FROM BOOK_DIM bd
        WHERE bd.bookID = s.bookID AND bd.copyID = s.copyID
    );

    v_count := SQL%ROWCOUNT;

    UPDATE BOOK_DIM bd
    SET (
        bookStatus,
        bookCondition
    ) = (
        SELECT
            s.clean_bookStatus,
            s.clean_bookCondition
        FROM BOOK_STAGING_V s
        WHERE s.bookID = bd.bookID AND s.copyID = bd.copyID
    )
    WHERE EXISTS (
        SELECT 1 FROM BOOK_STAGING_V s
        WHERE s.bookID = bd.bookID AND s.copyID = bd.copyID
        AND (bd.bookStatus != s.clean_bookStatus OR bd.bookCondition !=
s.clean_bookCondition)
    );

    v_updated := SQL%ROWCOUNT;

    SELECT COUNT(*) INTO v_errors
    FROM BOOK_STAGING_V s
```

Sample Assignment Report

```
    WHERE (s.title_cleaned = 'Y' OR s.imei_generated = 'Y' OR s.year_defaulted =
'Y')
      AND NOT EXISTS (
          SELECT 1 FROM BOOK_DIM bd
          WHERE bd.bookID = s.bookID AND bd.copyID = s.copyID
      );

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('BOOK_DIM incremental load completed:');
    DBMS_OUTPUT.PUT_LINE('  - New records inserted: ' || v_count);
    DBMS_OUTPUT.PUT_LINE('  - Existing records updated: ' || v_updated);
    DBMS_OUTPUT.PUT_LINE('  - Data quality corrections in new records: ' ||
v_errors);
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in BOOK_DIM incremental load: ' || SQLERRM);
        RAISE;
END;
/
```

MEMBER_DIM

```
CREATE OR REPLACE PROCEDURE LOAD_MEMBER_DIM_NEW_RECORDS
AS
    v_new_records NUMBER := 0;
    v_errors      NUMBER := 0;

    -- Cursor to select only members from staging that do not exist in the
dimension table
    CURSOR new_members_cursor IS
        SELECT s.*
        FROM MEMBER_STAGING_V s
        WHERE NOT EXISTS (
            SELECT 1
            FROM MEMBER_DIM md
            WHERE md.memberID = s.memberID
        );

BEGIN
    FOR rec IN new_members_cursor LOOP
        INSERT INTO MEMBER_DIM (
            member_key, memberID, memberName, memberEmail, dob, contactNum,
            memberCountry, memberState, memberCity, memberZipcode,
            effective_start_date, effective_end_date, is_current_flag
        ) VALUES (
            member_dim_seq.NEXTVAL, rec.memberID, rec.clean_memberName,
rec.clean_memberEmail,
            rec.clean_memberDob, rec.clean_contactNum, rec.clean_memberCountry,
rec.clean_memberState,
            rec.clean_memberCity, rec.clean_memberZipcode, SYSDATE, DATE
```

Sample Assignment Report

```
'9999-12-31', '1'
        );
        v_new_records := v_new_records + 1;
    END LOOP;

    SELECT COUNT(*) INTO v_errors
    FROM MEMBER_STAGING_V
    WHERE name_cleaned = 'Y' OR email_generated = 'Y' OR phone_defaulted = 'Y' OR
dob_defaulted = 'Y';

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('New members load completed for MEMBER_DIM:');
    DBMS_OUTPUT.PUT_LINE('  - New members inserted: ' || v_new_records);
    DBMS_OUTPUT.PUT_LINE('  - Data quality corrections in source: ' || v_errors);

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error during new members load for MEMBER_DIM: ' ||
SQLERRM);
        RAISE;
END LOAD_MEMBER_DIM_NEW_RECORDS;
/
```

DATE_DIM

```
CREATE OR REPLACE PROCEDURE LOAD_DATE_DIM_INCREMENTAL(p_end_year IN NUMBER
DEFAULT NULL) AS
    v_max_date DATE;
    v_target_date DATE;
    v_count NUMBER := 0;
BEGIN
    SELECT MAX(CAL_DATE) INTO v_max_date FROM DATE_DIM;

    v_target_date := v_max_date + 365;

    IF p_end_year IS NOT NULL AND p_end_year > 0 THEN
        v_target_date := TO_DATE(p_end_year || '-12-31', 'YYYY-MM-DD');
    END IF;

    INSERT INTO DATE_DIM (
        DATE_KEY, CAL_DATE, FULL_DESC, DAY_WEEK, DAY_NUM_MONTH, DAY_NUM_YEAR,
        LAST_DAY_IND, CAL_WEEK_END_DATE, CAL_WEEK_YEAR, CAL_MONTH_NAME,
        CAL_MONTH_YEAR, CAL_YEAR_MONTH, CAL_QUARTER, CAL_YEAR_QUARTER, CAL_YEAR,
        HOLIDAY_IND, WEEKDAY_IND, EXAM_WEEK_IND, STUDY_WEEK_IND, FESTIVE_EVENT
    )
    SELECT
        DATE_DIM_SEQ.NEXTVAL,
        v_date, full_desc, day_week, day_num_month, day_num_year,
        last_day_ind, cal_week_end_date, cal_week_year,
        TRIM(cal_month_name), -- Logic to scrub dirty data
```

Sample Assignment Report

```
        cal_month_year, cal_year_month, cal_quarter, cal_year_quarter, cal_year,
        holiday_ind, weekday_ind, exam_week_ind, study_week_ind, festive_event
    FROM DATE_STAGING_V
    WHERE v_date > v_max_date
      AND v_date <= v_target_date;

    v_count := SQL%ROWCOUNT;

    UPDATE DATE_DIM
    SET STUDY_WEEK_IND = 'Y'
    WHERE CAL_DATE > v_max_date
      AND ((TO_CHAR(CAL_DATE, 'MM-DD') >= '01-06' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '01-12')
        OR (TO_CHAR(CAL_DATE, 'MM-DD') >= '05-19' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '05-25')
        OR (TO_CHAR(CAL_DATE, 'MM-DD') >= '09-29' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '10-05'));

    UPDATE DATE_DIM
    SET EXAM_WEEK_IND = 'Y'
    WHERE CAL_DATE > v_max_date
      AND ((TO_CHAR(CAL_DATE, 'MM-DD') >= '01-13' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '01-19')
        OR (TO_CHAR(CAL_DATE, 'MM-DD') >= '05-26' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '06-08')
        OR (TO_CHAR(CAL_DATE, 'MM-DD') >= '10-06' AND TO_CHAR(CAL_DATE, 'MM-DD')
<= '10-19'));

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('DATE_DIM incremental load completed: ' || v_count || '
new records inserted.');
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in DATE_DIM incremental load: ' || SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

STAFF_DIM

```
CREATE OR REPLACE PROCEDURE LOAD_STAFF_DIM_NEW_RECORDS
AS
    v_new_records NUMBER := 0;
    v_errors      NUMBER := 0;

    -- Cursor to select only staff from staging who do not exist in the dimension
table
    CURSOR new_staff_cursor IS
        SELECT s.*
        FROM STAFF_STAGING_V s
```

```
        WHERE NOT EXISTS (
            SELECT 1
            FROM STAFF_DIM sd
            WHERE sd.staffID = s.staffID
        );

BEGIN
    FOR rec IN new_staff_cursor LOOP
        INSERT INTO STAFF_DIM (
            staff_key, staffID, staffName, staffEmail, staffPhoneNum,
            department, position, hireDate,
            effective_start_date, effective_end_date, is_current_flag
        ) VALUES (
            staff_dim_seq.NEXTVAL, rec.staffID, rec.clean_staffName,
rec.clean_staffEmail,
            rec.clean_staffPhoneNum, rec.clean_department, rec.clean_position,
            rec.clean_hireDate, SYSDATE, DATE '9999-12-31', '1'
        );
        v_new_records := v_new_records + 1;
    END LOOP;

    SELECT COUNT(*) INTO v_errors
    FROM STAFF_STAGING_V
    WHERE name_cleaned = 'Y' OR email_generated = 'Y' OR phone_defaulted = 'Y' OR
date_corrected = 'Y';

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('New staff load completed for STAFF_DIM:');
    DBMS_OUTPUT.PUT_LINE('  - New staff inserted: ' || v_new_records);
    DBMS_OUTPUT.PUT_LINE('  - Data quality corrections in source: ' || v_errors);

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error during new staff load for STAFF_DIM: ' ||
SQLERRM);
        RAISE;
END LOAD_STAFF_DIM_NEW_RECORDS;
/
```

Sample Assignment Report

SUPPLIER_DIM

```
CREATE OR REPLACE PROCEDURE LOAD_SUPPLIER_DIM_NEW_RECORDS
AS
    v_new_records NUMBER := 0;
    v_errors      NUMBER := 0;

    -- Cursor to select only suppliers from staging that do not exist in the
dimension table
    CURSOR new_suppliers_cursor IS
        SELECT s.*
```

```
        FROM SUPPLIER_STAGING_V s
        WHERE NOT EXISTS (
            SELECT 1
            FROM SUPPLIER_DIM sd
            WHERE sd.supplierID = s.supplierID
        );

BEGIN
    FOR rec IN new_suppliers_cursor LOOP
        INSERT INTO SUPPLIER_DIM (
            supplier_key, supplierID, supplierName, supplierLocation,
            supplierContactNum, supplierEmail,
            effective_start_date, effective_end_date, is_current_flag
        ) VALUES (
            supplier_dim_seq.NEXTVAL, rec.supplierID, rec.clean_supplierName,
            rec.clean_supplierLocation, rec.clean_supplierContactNum,
rec.clean_supplierEmail,
            SYSDATE, DATE '9999-12-31', '1'
        );
        v_new_records := v_new_records + 1;
    END LOOP;

    SELECT COUNT(*) INTO v_errors
    FROM SUPPLIER_STAGING_V
    WHERE name_cleaned = 'Y' OR location_defaulted = 'Y' OR phone_defaulted = 'Y'
OR email_generated = 'Y';

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('New suppliers load completed for SUPPLIER_DIM:');
    DBMS_OUTPUT.PUT_LINE('  - New suppliers inserted: ' || v_new_records);
    DBMS_OUTPUT.PUT_LINE('  - Data quality corrections in source: ' || v_errors);

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error during new suppliers load for SUPPLIER_DIM: '
|| SQLERRM);
        RAISE;
END LOAD_SUPPLIER_DIM_NEW_RECORDS;
/
```

Sample Assignment Report

RESERVATION_FACT

```
CREATE OR REPLACE PROCEDURE LOAD_RESERVATION_FACT_INC(p_load_date IN DATE DEFAULT
SYSDATE) AS
    v_count NUMBER := 0;
    v_updated NUMBER := 0;
    v_errors NUMBER := 0;
BEGIN
    INSERT INTO RESERVATION_FACT (
        member_key, book_key, staff_key, reserve_start_date_key,
```

```
        reserve_end_date_key, reserveID, reservationStatus, reservationDuration
    )
    SELECT
        M.member_key,
        B.book_key,
        S.staff_key,
        D_START.date_key AS reserve_start_date_key,
        D_END.date_key AS reserve_end_date_key,
        rs.reserveID,
        rs.clean_reserveStatus,
        rs.clean_reservationDuration
    FROM RESERVATION_STAGING_V rs
    JOIN DATE_DIM D_START ON rs.reserveStartDate = D_START.cal_date
    JOIN DATE_DIM D_END ON rs.clean_reserveEndDate = D_END.cal_date
    JOIN MEMBER_DIM M ON rs.memberID = M.memberID AND M.is_current_flag = '1'
    JOIN BOOK_DIM B ON rs.copyID = B.copyID
    JOIN STAFF_DIM S ON rs.staffID = S.staffID AND S.is_current_flag = '1'
    WHERE NOT EXISTS (SELECT 1 FROM RESERVATION_FACT rf WHERE rf.reserveID =
rs.reserveID)
      AND rs.reserveStartDate >= p_load_date - 1;

    v_count := SQL%ROWCOUNT;

    UPDATE RESERVATION_FACT rf
    SET reservationStatus = (
        SELECT rs.clean_reserveStatus
        FROM RESERVATION_STAGING_V rs
        WHERE rs.reserveID = rf.reserveID
    )
    WHERE EXISTS (
        SELECT 1 FROM RESERVATION_STAGING_V rs
        WHERE rs.reserveID = rf.reserveID
        AND rf.reservationStatus != rs.clean_reserveStatus
    );

    v_updated := SQL%ROWCOUNT;

    SELECT COUNT(*) INTO v_errors
    FROM RESERVATION_STAGING_V rs
    WHERE (rs.enddate_generated = 'Y' OR rs.status_defaulted = 'Y' OR
rs.duration_fixed = 'Y')
      AND rs.reserveStartDate >= p_load_date - 1
      AND NOT EXISTS (SELECT 1 FROM RESERVATION_FACT rf WHERE rf.reserveID =
rs.reserveID);

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('RESERVATION_FACT incremental load completed:');
    DBMS_OUTPUT.PUT_LINE('  - New records inserted: ' || v_count);
    DBMS_OUTPUT.PUT_LINE('  - Existing records updated: ' || v_updated);
    DBMS_OUTPUT.PUT_LINE('  - Data quality corrections: ' || v_errors);
EXCEPTION
    WHEN OTHERS THEN
```

Sample Assignment Report

```
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in RESERVATION_FACT incremental load: ' ||
SQLERRM);
        RAISE;
END;
/
```

LOAN_FACT

```
CREATE OR REPLACE PROCEDURE LOAD_LOAN_FACT_INCREMENTAL(p_load_date IN DATE
DEFAULT SYSDATE) AS
    v_count NUMBER := 0;
    v_updated NUMBER := 0;
    v_errors NUMBER := 0;
BEGIN
    INSERT INTO LOAN_FACT (
        member_key, book_key, staff_key, date_key, loanID,
        loanDuration, overdueDays, loanStatus, totalFine, fine_paid_flag
    )
    SELECT
        M.member_key, B.book_key, S.staff_key, D.date_key, ls.loanID,
        ls.clean_loanDuration, ls.clean_overdueDays, ls.clean_loanStatus,
        COALESCE(f.totalFineAmt, 0) AS totalFine, f.paid_flag AS fine_paid_flag
    FROM LOAN_STAGING_V ls
    JOIN DATE_DIM D ON ls.loanDate = D.cal_date
    JOIN MEMBER_DIM M ON ls.memberID = M.memberID AND M.is_current_flag = '1'
    JOIN BOOK_DIM B ON ls.copyID = B.copyID
    JOIN STAFF_DIM S ON ls.staffID = S.staffID AND S.is_current_flag = '1'
    LEFT JOIN FINE_SUMMARY_V f ON ls.fine_lookup_id = f.loanID
    WHERE NOT EXISTS (SELECT 1 FROM LOAN_FACT lf WHERE lf.loanID = ls.loanID)
      AND ls.loanDate >= p_load_date - 1;

    v_count := SQL%ROWCOUNT;

    UPDATE LOAN_FACT lf
    SET (overdueDays, loanStatus, totalFine, fine_paid_flag) =
    (
        SELECT
            ls.clean_overdueDays,
            ls.clean_loanStatus,
            COALESCE(f.totalFineAmt, 0),
            f.paid_flag
        FROM LOAN_STAGING_V ls
        LEFT JOIN FINE_SUMMARY_V f ON ls.fine_lookup_id = f.loanID
        WHERE ls.loanID = lf.loanID
    )
    WHERE EXISTS (
        SELECT 1 FROM LOAN_STAGING_V ls
        LEFT JOIN FINE_SUMMARY_V f ON ls.fine_lookup_id = f.loanID
        WHERE ls.loanID = lf.loanID
        AND (lf.loanStatus != ls.clean_loanStatus
```

Sample Assignment Report

```
            OR NVL(lf.overdueDays, -1) != NVL(ls.clean_overdueDays, -1)
            OR NVL(lf.totalFine, -1) != NVL(COALESCE(f.totalFineAmt, 0), -1)
            OR NVL(lf.fine_paid_flag, -1) != NVL(f.paid_flag, -1))
    );

    v_updated := SQL%ROWCOUNT;

    SELECT COUNT(*) INTO v_errors
    FROM LOAN_STAGING_V ls
    WHERE (ls.duration_corrected = 'Y' OR ls.status_defaulted = 'Y')
      AND ls.loanDate >= p_load_date - 1;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('LOAN_FACT incremental load completed:');
    DBMS_OUTPUT.PUT_LINE('  - New records inserted: ' || v_count);
    DBMS_OUTPUT.PUT_LINE('  - Existing records updated: ' || v_updated);
    DBMS_OUTPUT.PUT_LINE('  - Data quality corrections: ' || v_errors);
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in LOAN_FACT incremental load: ' || SQLERRM);
        RAISE;
END;
/
```

PURCHASE_FACT

```
CREATE OR REPLACE PROCEDURE LOAD_PURCHASE_FACT_INCREMENTAL(p_load_date IN DATE
DEFAULT SYSDATE) AS
    v_count NUMBER := 0;
    v_errors NUMBER := 0;
    v_duplicates NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_duplicates
    FROM PURCHASE_STAGING_V ps
    WHERE EXISTS (SELECT 1 FROM PURCHASE_FACT pf WHERE pf.purchaseID =
ps.purchaseID)
      AND ps.purchaseDate >= p_load_date - 1;

    -- FIX: Applying the same de-duplication logic to the incremental load.
    INSERT INTO PURCHASE_FACT (
        date_key, book_key, staff_key, supplier_key, purchaseID,
        purchaseQuantity, purchaseUnitCost, purchaseTotalCost
    )
    WITH BOOK_DIM_DEDUPED AS (
        SELECT
            bookID,
            MIN(book_key) as book_key
        FROM BOOK_DIM
        GROUP BY bookID
    )
```

Sample Assignment Report

```
    SELECT
        D.date_key,
        B.book_key,
        S.staff_key,
        SUP.supplier_key,
        ps.purchaseID,
        ps.clean_purchaseQty,
        ps.clean_purchaseUnitCost,
        (ps.clean_purchaseQty * ps.clean_purchaseUnitCost) AS purchaseTotalCost
    FROM PURCHASE_STAGING_V ps
    JOIN DATE_DIM D ON ps.purchaseDate = D.cal_date
    JOIN BOOK_DIM_DEDUPED B ON ps.bookID = B.bookID -- Corrected join
    JOIN STAFF_DIM S ON ps.staffID = S.staffID AND S.is_current_flag = '1'
    JOIN SUPPLIER_DIM SUP ON ps.supplierID = SUP.supplierID AND
SUP.is_current_flag = '1'
    WHERE NOT EXISTS (SELECT 1 FROM PURCHASE_FACT pf WHERE pf.purchaseID =
ps.purchaseID)
      AND ps.purchaseDate >= p_load_date - 1;

    v_count := SQL%ROWCOUNT;

    SELECT COUNT(*) INTO v_errors
    FROM PURCHASE_STAGING_V ps
    WHERE (ps.qty_corrected = 'Y' OR ps.cost_corrected = 'Y')
      AND NOT EXISTS (SELECT 1 FROM PURCHASE_FACT pf WHERE pf.purchaseID =
ps.purchaseID)
      AND ps.purchaseDate >= p_load_date - 1;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('PURCHASE_FACT incremental load completed:');
    DBMS_OUTPUT.PUT_LINE('  - New records inserted: ' || v_count);
    DBMS_OUTPUT.PUT_LINE('  - Data quality corrections in new records: ' ||
v_errors);
    DBMS_OUTPUT.PUT_LINE('  - Duplicate records skipped: ' || v_duplicates);
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in PURCHASE_FACT incremental load: ' ||
SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

SALES_FACT

```
CREATE OR REPLACE PROCEDURE LOAD_SALES_FACT_INCREMENTAL(p_load_date IN DATE
DEFAULT SYSDATE) AS
    v_count NUMBER := 0;
    v_errors NUMBER := 0;
    v_duplicates NUMBER := 0;
BEGIN
```

```
    SELECT COUNT(*) INTO v_duplicates
    FROM SALES_STAGING_V ss
    WHERE EXISTS (
        SELECT 1 FROM SALES_FACT sf
        WHERE sf.orderID = ss.orderID
          AND EXISTS (SELECT 1 FROM BOOK_DIM B WHERE B.book_key = sf.book_key AND
B.bookID = ss.bookID)
    );

    -- Insert new records with validation
    -- FIX: Applying the same de-duplication logic to the incremental load.
    INSERT INTO SALES_FACT (
        date_key, member_key, book_key, staff_key, orderID,
        orderQty, orderUnitPrice, orderTotalPrice
    )
    WITH BOOK_DIM_DEDUPED AS (
        SELECT
            bookID,
            MIN(book_key) as book_key
        FROM BOOK_DIM
        GROUP BY bookID
    )
    SELECT
        D.date_key,
        M.member_key,
        B.book_key,
        S.staff_key,
        ss.orderID,
        ss.clean_orderQty,
        ss.clean_orderUnitPrice,
        (ss.clean_orderQty * ss.clean_orderUnitPrice) AS orderTotalPrice
    FROM SALES_STAGING_V ss
    JOIN DATE_DIM D ON ss.orderDate = D.cal_date
    JOIN MEMBER_DIM M ON ss.memberID = M.memberID AND M.is_current_flag = '1'
    JOIN BOOK_DIM_DEDUPED B ON ss.bookID = B.bookID -- Corrected join
    JOIN STAFF_DIM S ON ss.staffID = S.staffID AND S.is_current_flag = '1'
    WHERE NOT EXISTS (
        SELECT 1 FROM SALES_FACT sf
        WHERE sf.orderID = ss.orderID
          AND EXISTS (SELECT 1 FROM BOOK_DIM B2 WHERE B2.book_key = sf.book_key
AND B2.bookID = ss.bookID)

    )
    AND ss.orderDate >= p_load_date - 1;

    v_count := SQL%ROWCOUNT;

    SELECT COUNT(*) INTO v_errors
    FROM SALES_STAGING_V ss
    WHERE (ss.qty_corrected = 'Y' OR ss.price_corrected = 'Y')
      AND ss.orderDate >= p_load_date - 1
      AND NOT EXISTS (
```

Sample Assignment Report

```
          SELECT 1 FROM SALES_FACT sf
          WHERE sf.orderID = ss.orderID
            AND EXISTS (SELECT 1 FROM BOOK_DIM B WHERE B.book_key = sf.book_key
AND B.bookID = ss.bookID)
      );

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('SALES_FACT incremental load completed:');
    DBMS_OUTPUT.PUT_LINE('  - New records inserted: ' || v_count);
    DBMS_OUTPUT.PUT_LINE('  - Data quality corrections in new records: ' ||
v_errors);
    DBMS_OUTPUT.PUT_LINE('  - Duplicate records skipped: ' || v_duplicates);
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error in SALES_FACT incremental load: ' ||
SQLERRM);
        RAISE;
END;
/
```

Sample Assignment Report

2.3 Script for Type 2 SCD Maintenance

MEMBER_DIM

```
CREATE OR REPLACE PROCEDURE MAINTAIN_MEMBER_DIM_SCD2
AS
    v_updated_records NUMBER := 0;

    -- Cursor to find existing, current members whose attributes have changed
    CURSOR changed_members_cursor IS
        SELECT
            s.*,
            md.member_key
        FROM
            MEMBER_STAGING_V s
        JOIN
            MEMBER_DIM md ON s.memberID = md.memberID
        WHERE
            md.is_current_flag = '1'  -- Only compare against the current version
            AND (
                -- Compare each attribute, handling NULLs gracefully
                NVL(md.memberName, 'X') != NVL(s.clean_memberName, 'X') OR
                NVL(md.memberEmail, 'X') != NVL(s.clean_memberEmail, 'X') OR
                    NVL(md.dob, DATE '1900-01-01') != NVL(s.clean_memberDob, DATE
'1900-01-01') OR
                NVL(md.contactNum, 'X') != NVL(s.clean_contactNum, 'X') OR
                NVL(md.memberCountry, 'X') != NVL(s.clean_memberCountry, 'X') OR
                NVL(md.memberState, 'X') != NVL(s.clean_memberState, 'X') OR
                NVL(md.memberCity, 'X') != NVL(s.clean_memberCity, 'X') OR
                NVL(md.memberZipcode, 'X') != NVL(s.clean_memberZipcode, 'X')
            );

BEGIN
    -- Loop through each member with detected changes
    FOR rec IN changed_members_cursor LOOP
        -- Step 1: Expire the old (current) record
        UPDATE MEMBER_DIM
        SET
            effective_end_date = SYSDATE - 1,
            is_current_flag = '0'
        WHERE
            member_key = rec.member_key;

        -- Step 2: Insert the new record with updated information
        INSERT INTO MEMBER_DIM (
            member_key, memberID, memberName, memberEmail, dob, contactNum,
            memberCountry, memberState, memberCity, memberZipcode,
            effective_start_date, effective_end_date, is_current_flag
        ) VALUES (
                        member_dim_seq.NEXTVAL,  rec.memberID,  rec.clean_memberName,
```

Sample Assignment Report

```
rec.clean_memberEmail,
               rec.clean_memberDob, rec.clean_contactNum, rec.clean_memberCountry,
rec.clean_memberState,
                    rec.clean_memberCity,  rec.clean_memberZipcode,  SYSDATE,  DATE
'9999-12-31', '1'
        );
        v_updated_records := v_updated_records + 1;
    END LOOP;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('MEMBER_DIM SCD2 maintenance completed:');
    DBMS_OUTPUT.PUT_LINE('  - Member changes processed: ' || v_updated_records);

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
            DBMS_OUTPUT.PUT_LINE('Error during MEMBER_DIM SCD2 maintenance: ' ||
SQLERRM);
        RAISE;
END MAINTAIN_MEMBER_DIM_SCD2;
/
```

STAFF_DIM

```
CREATE OR REPLACE PROCEDURE MAINTAIN_STAFF_DIM_SCD2
AS
    v_updated_records NUMBER := 0;

    -- Cursor to find existing, current staff whose attributes have changed
    CURSOR changed_staff_cursor IS
        SELECT
            s.*,
            sd.staff_key
        FROM
            STAFF_STAGING_V s
        JOIN
            STAFF_DIM sd ON s.staffID = sd.staffID
        WHERE
            sd.is_current_flag = '1'  -- Only compare against the current version
            AND (
                NVL(sd.staffName, 'X') != NVL(s.clean_staffName, 'X') OR
                NVL(sd.staffEmail, 'X') != NVL(s.clean_staffEmail, 'X') OR
                NVL(sd.staffPhoneNum, 'X') != NVL(s.clean_staffPhoneNum, 'X') OR
                NVL(sd.department, 'X') != NVL(s.clean_department, 'X') OR
                NVL(sd.position, 'X') != NVL(s.clean_position, 'X')
            );

BEGIN
    FOR rec IN changed_staff_cursor LOOP
        UPDATE STAFF_DIM
        SET
```

Sample Assignment Report

```
            effective_end_date = SYSDATE - 1,
            is_current_flag = '0'
        WHERE
            staff_key = rec.staff_key;

        INSERT INTO STAFF_DIM (
            staff_key, staffID, staffName, staffEmail, staffPhoneNum,
            department, position, hireDate,
            effective_start_date, effective_end_date, is_current_flag
        ) VALUES (
                        staff_dim_seq.NEXTVAL,  rec.staffID,  rec.clean_staffName,
rec.clean_staffEmail,
            rec.clean_staffPhoneNum, rec.clean_department, rec.clean_position,
            rec.clean_hireDate, SYSDATE, DATE '9999-12-31', '1'
        );
        v_updated_records := v_updated_records + 1;
    END LOOP;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('STAFF_DIM SCD2 maintenance completed:');
    DBMS_OUTPUT.PUT_LINE('  - Staff changes processed: ' || v_updated_records);

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
            DBMS_OUTPUT.PUT_LINE('Error during STAFF_DIM SCD2 maintenance: ' ||
SQLERRM);
        RAISE;
END MAINTAIN_STAFF_DIM_SCD2;
/
```

Sample Assignment Report

SUPPLIER_DIM

```
CREATE OR REPLACE PROCEDURE MAINTAIN_SUPPLIER_DIM_SCD2
AS
    v_updated_records NUMBER := 0;

    -- Cursor to find existing, current suppliers whose attributes have changed
    CURSOR changed_suppliers_cursor IS
        SELECT
            s.*,
            sd.supplier_key
        FROM
            SUPPLIER_STAGING_V s
        JOIN
            SUPPLIER_DIM sd ON s.supplierID = sd.supplierID
        WHERE
            sd.is_current_flag = '1'  -- Only compare against the current version
            AND (
                NVL(sd.supplierName, 'X') != NVL(s.clean_supplierName, 'X') OR
                    NVL(sd.supplierLocation, 'X') != NVL(s.clean_supplierLocation,
```

```
'X') OR
                                                NVL(sd.supplierContactNum,   'X')   !=
NVL(s.clean_supplierContactNum, 'X') OR
                NVL(sd.supplierEmail, 'X') != NVL(s.clean_supplierEmail, 'X')
            );

BEGIN
    FOR rec IN changed_suppliers_cursor LOOP
        UPDATE SUPPLIER_DIM
        SET
            effective_end_date = SYSDATE - 1,
            is_current_flag = '0'
        WHERE
            supplier_key = rec.supplier_key;

        INSERT INTO SUPPLIER_DIM (
            supplier_key, supplierID, supplierName, supplierLocation,
            supplierContactNum, supplierEmail,
            effective_start_date, effective_end_date, is_current_flag
        ) VALUES (
            supplier_dim_seq.NEXTVAL, rec.supplierID, rec.clean_supplierName,
                        rec.clean_supplierLocation, rec.clean_supplierContactNum,
rec.clean_supplierEmail,
            SYSDATE, DATE '9999-12-31', '1'
        );
        v_updated_records := v_updated_records + 1;
    END LOOP;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('SUPPLIER_DIM SCD2 maintenance completed:');
            DBMS_OUTPUT.PUT_LINE('    -   Supplier   changes   processed:   '   ||
v_updated_records);

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
           DBMS_OUTPUT.PUT_LINE('Error during SUPPLIER_DIM SCD2 maintenance: ' ||
SQLERRM);
        RAISE;
END MAINTAIN_SUPPLIER_DIM_SCD2;
/
```

Sample Assignment Report

Chapter 3: Business Analytics Report

3.1 Leong Kah Tian

3.1.1 Sales Trend & Seasonality Analysis by Book Category

SQL Code:

```
SET PAGESIZE 35
SET LINESIZE 130

SET VERIFY OFF
SET ECHO OFF
TTITLE OFF
CLEAR COLUMNS
CLEAR BREAK

ACCEPT report_year PROMPT 'Enter the report year (e.g., 2025): '

-- fixed metric label
COLUMN METRIC_LABEL NOPRINT NEW_VALUE METRIC_LABEL
SELECT 'Loans' AS METRIC_LABEL FROM dual;

COLUMN RUN_DT NOPRINT NEW_VALUE RUN_DT
SELECT TO_CHAR(SYSDATE, 'DD-MON-YY') AS RUN_DT FROM dual;

COLUMN Q1 NOPRINT NEW_VALUE Q1
COLUMN Q2 NOPRINT NEW_VALUE Q2
COLUMN Q3 NOPRINT NEW_VALUE Q3
COLUMN Q4 NOPRINT NEW_VALUE Q4

SELECT
  '&report_year-Q1' AS Q1, '&report_year-Q2' AS Q2,
  '&report_year-Q3' AS Q3, '&report_year-Q4' AS Q4
FROM dual;

--Column Formatting
COLUMN RANK_POP         HEADING 'RANK'                  FORMAT 999,99
COLUMN CATEGORY         HEADING 'CATEGORY'              FORMAT A14
COLUMN Q_1              HEADING '&Q1'                   FORMAT 999,999,999
COLUMN Q_2              HEADING '&Q2'                   FORMAT 999,999,999
COLUMN Q_3              HEADING '&Q3'                   FORMAT 999,999,999
COLUMN Q_4              HEADING '&Q4'                   FORMAT 999,999,999
COLUMN T12M             HEADING 'TOTAL|QTY|&report_year'  FORMAT 999,999,990
COLUMN PEAK_QTR         HEADING 'Peak|Quarter'          FORMAT A8
COLUMN YOY_GROWTH_PCT   HEADING 'Growth %'          FORMAT 990.0
COLUMN METRIC_TOT       HEADING 'TOTAL|&METRIC_LABEL'   FORMAT 999,999,990
```

Sample Assignment Report

```
COLUMN POPULARITY_PCT  HEADING 'Popularity|Share %'    FORMAT 990.0

BREAK ON REPORT
COMPUTE SUM LABEL 'TOTAL' OF Q_1 Q_2 Q_3 Q_4 T12M METRIC_TOT ON REPORT

TTITLE ON
TTITLE
CENTER'+-------------------------------------------------------------------
-------------------------------------------------------+' SKIP 1 -
       CENTER '                              QUARTERLY SALES TREND BY BOOK
CATEGORY (Year: &report_year)                                ' SKIP 1 -

CENTER'+-------------------------------------------------------------------
-------------------------------------------------------+' SKIP 1 -
       LEFT 'DATE: &RUN_DT' SKIP 1 -
       LEFT 'PAGE:' SQL.PNO SKIP 1

--Data Preparation & Final Output
WITH report_period AS (
  SELECT TO_DATE('01-01-&report_year','DD-MM-YYYY') AS start_of_year,
         TO_DATE('31-12-&report_year','DD-MM-YYYY') AS end_of_year
  FROM dual
),
-- Quarterly sales qty by category
current_year AS (
  SELECT NVL(bd.bookCategory,'UNKNOWN') AS category,
         dd.cal_quarter,
         SUM(sf.orderQty)              AS qty
  FROM sales_fact sf
  JOIN date_dim dd ON dd.date_key = sf.date_key
  JOIN book_dim bd ON bd.book_key = sf.book_key
  JOIN report_period rp
    ON dd.cal_date BETWEEN rp.start_of_year AND rp.end_of_year
  GROUP BY NVL(bd.bookCategory,'UNKNOWN'), dd.cal_quarter
),
prev_year AS (
  SELECT NVL(bd.bookCategory,'UNKNOWN') AS category,
         SUM(sf.orderQty)               AS qty_prev12
  FROM sales_fact sf
  JOIN date_dim dd ON dd.date_key = sf.date_key
  JOIN book_dim bd ON bd.book_key = sf.book_key
  JOIN report_period rp
    ON dd.cal_date BETWEEN ADD_MONTHS(rp.start_of_year,-12) AND
ADD_MONTHS(rp.end_of_year,-12)
  GROUP BY NVL(bd.bookCategory,'UNKNOWN')
),
loan_year AS (
  SELECT NVL(bd.bookCategory,'UNKNOWN') AS category,
         COUNT(DISTINCT lf.loanID)      AS loans_tot
```

Sample Assignment Report

```
  FROM loan_fact lf
  JOIN date_dim dd ON dd.date_key = lf.date_key
  JOIN book_dim bd ON bd.book_key = lf.book_key
  JOIN report_period rp
    ON dd.cal_date BETWEEN rp.start_of_year AND rp.end_of_year
  GROUP BY NVL(bd.bookCategory,'UNKNOWN')
),
pivoted_data AS (
  SELECT
    c.category,
    SUM(CASE WHEN c.cal_quarter = 1 THEN c.qty ELSE 0 END) AS Q_1,
    SUM(CASE WHEN c.cal_quarter = 2 THEN c.qty ELSE 0 END) AS Q_2,
    SUM(CASE WHEN c.cal_quarter = 3 THEN c.qty ELSE 0 END) AS Q_3,
    SUM(CASE WHEN c.cal_quarter = 4 THEN c.qty ELSE 0 END) AS Q_4,
    SUM(c.qty) AS T12M,
    MAX(c.cal_quarter) KEEP (DENSE_RANK FIRST ORDER BY c.qty DESC,
c.cal_quarter ASC) AS peak_qtr
  FROM current_year c
  GROUP BY c.category
),
base_rows AS (
  SELECT
    p.category,
    p.Q_1, p.Q_2, p.Q_3, p.Q_4,
    p.T12M,
    'Q' || p.peak_qtr AS peak_qtr,
    ROUND( (p.T12M - NVL(py.qty_prev12,0)) / NULLIF(py.qty_prev12,0) * 100,
1) AS yoy_growth_pct,
    NVL(ly.loans_tot,0) AS metric_tot,
    (p.T12M + NVL(ly.loans_tot,0)) AS pop_val
  FROM pivoted_data p
  LEFT JOIN prev_year py ON py.category = p.category
  LEFT JOIN loan_year ly ON ly.category = p.category
)
SELECT
  RANK() OVER (ORDER BY b.pop_val DESC) AS RANK_POP,
  b.category,
  b.Q_1,
  b.Q_2,
  b.Q_3,
  b.Q_4,
  b.T12M,
  b.peak_qtr,
  b.yoy_growth_pct,
  b.metric_tot,
  ROUND( b.pop_val / NULLIF(SUM(b.pop_val) OVER (), 0) * 100, 1) AS
popularity_pct
FROM base_rows b
ORDER BY popularity_pct DESC, b.category;
```

Sample Assignment Report

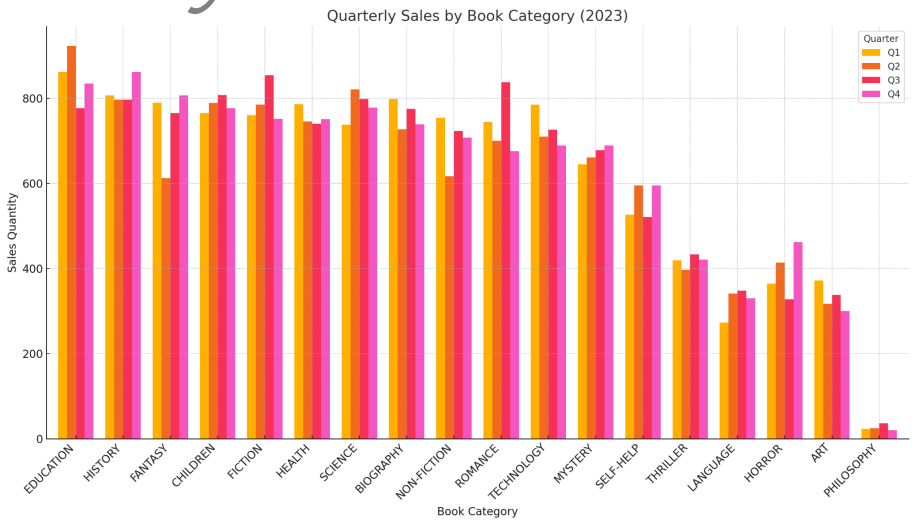
```
TTITLE OFF
CLEAR COLUMNS
CLEAR BREAK
```

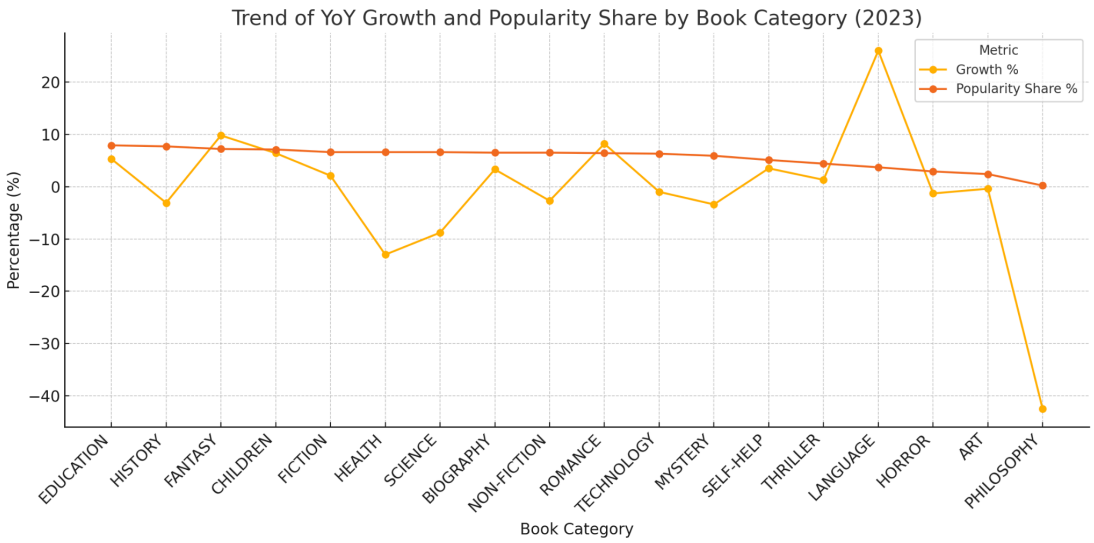
Output:

```
+-------------------------------------------------------------------------------------------------------------------------------------+
                                        QUARTERLY SALES TREND BY BOOK CATEGORY (Year: 2023)
+-------------------------------------------------------------------------------------------------------------------------------------+
DATE: 08-SEP-25
PAGE:         1
                                                                                   TOTAL
                                                                                     QTY Peak                          TOTAL Popularity
  RANK CATEGORY           2023-Q1      2023-Q2      2023-Q3      2023-Q4          2023 Quarter  Growth %          Loans    Share %
------- -------------- ------------ ------------ ------------ ------------ ------------ -------- -------- ------------ ----------
      1 EDUCATION               862          923          777          835        3,397 Q2            5.3          897        7.9
      2 HISTORY                 807          797          797          862        3,263 Q4           -3.1          940        7.7
      3 FANTASY                 790          612          765          807        2,974 Q4            9.8          934        7.2
      4 CHILDREN                765          789          808          777        3,139 Q3            6.4          738        7.1
      5 FICTION                 760          785          854          752        3,151 Q3            2.1          482        6.6
      7 HEALTH                  786          746          740          751        3,023 Q1          -13.0          571        6.6
      6 SCIENCE                 738          821          798          778        3,135 Q2           -8.8          493        6.6
      8 BIOGRAPHY               798          727          775          739        3,039 Q1            3.3          522        6.5
      9 NON-FICTION             754          617          723          708        2,802 Q1           -2.7          751        6.5
     10 ROMANCE                 744          700          838          676        2,958 Q3            8.2          534        6.4
     11 TECHNOLOGY              785          710          726          689        2,910 Q1           -1.0          525        6.3
     12 MYSTERY                 645          661          678          689        2,673 Q4           -3.4          561        5.9
     13 SELF-HELP               526          595          521          595        2,237 Q2            3.5          568        5.1
     14 THRILLER                419          397          433          421        1,670 Q3            1.3          743        4.4
     15 LANGUAGE                273          341          348          330        1,292 Q3           26.0          715        3.7
     16 HORROR                  364          414          328          462        1,568 Q4           -1.3            0        2.9
     17 ART                     372          317          338          300        1,327 Q1           -0.4            0        2.4
     18 PHILOSOPHY               23           25           36           20          104 Q3          -42.5            0        0.2
                           ------------ ------------ ------------ ------------ ------------                  ------------
TOTAL                         11,211       10,977       11,283       11,191       44,662                            9,974

18 rows selected.
```

Sample Assignment Report

Visualisation:





Justification:

Sample Assignment Report

This report analyses 2023 quarterly sales trends by book category to uncover seasonal patterns, highlight year-over-year (YoY) growth or decline, and guide category-level purchase decisions. The table's columns serve clear analytical purposes. Q1 to Q4 report the total order quantity for each quarter, helping to observe quarterly demand shifts and seasonal spikes. Peak Quarter identifies the quarter with the highest sales per category, signaling the best time for bulk ordering. Growth rate is calculated by subtracting 2022 total sales from 2023 total sales, dividing by 2022 total sales, and multiplying by 100, to show YoY performance changes. Loans Total reflects total borrowings in 2023, capturing reader demand from a non-purchase angle. Popularity Share rate is derived by adding 2023 sales and 2023 loans for the category, dividing by the total sum across all categories, then multiplying by 100. This shows the category's weight in the overall mix. Rank is based on combined sales and loan activity to prioritize importance across categories.

The data tells us that Education, History, and Fantasy are top-performing categories, each contributing a high share to total transactions and showing strong seasonal demand, particularly in Q2 and Q3, likely due to school semesters, textbook usage, and exam periods. In contrast, Philosophy and Art sit at the bottom of the ranking with negative growth rates like Philosophy is -42.5%, low loan activity, and weak quarterly demand. This suggesting low interest or outdated stock. This is significant because categories with high Popularity Share rate support faster stock turnover and should receive greater budget allocation, while those with sustained decline risk becoming obsolete and tying up inventory space.

The reason this happened is due to cyclical academic calendars and member preferences, where educational demand spikes mid-year, while niche or hobbyist categories lag. The evidence is visible in both the Quarterly Sales Chart and the YoY Growth versus Popularity Line Graph, where clear patterns of volume concentration emerge.

In the future, the library should adopt a data-driven purchasing strategy by prioritizing reorders for top-growth categories such as Education, Fantasy, and Children, placing bulk orders 4–6 weeks before their Peak Quarter, especially in Q2 and Q3, and increasing purchase quantities by approximately 25% year-over-year to meet strong and proven demand. For instance, if the Children's Books category shows a consistent peak in June, bulk orders

should be placed by mid-April. Conversely, library staff should reduce orders or shift to purchase-on-demand for low-growth, low-popularity categories like Art and Philosophy, particularly when Growth % remains negative for two consecutive years and total sales fall below the median, signaling weak demand. To boost borrowing in mid-tier categories such as Science, Romance, and Non-Fiction, the library can organize targeted promotional campaigns like “Category Days” or mini “Book Fairs” held one month prior to each category’s typical peak. For example, running a Romance Book Fair in January if it peaks in February. During the budgeting process, Growth % and Popularity Share % should be used as core filters to classify each category into one of three actionable groups: expand, sustain, or sunset. By aligning future inventory and promotional efforts with actual seasonality patterns and historical category performance, the library will be able to maximize member engagement, reduce overstock risk, and ensure the budget is spent where it yields the highest return.

3.1.2 Member Acquisition & Retention Trend Analysis

SQL Code:

Sample Assignment Report

```
SET PAGESIZE 35
SET LINESIZE 110
SET VERIFY OFF
SET FEEDBACK OFF
SET DEFINE OFF

--REPORT TITLE
TTITLE ON
TTITLE LEFT
'+-------------------------------------------------------------------------
-------------------------------------------------------------+' SKIP 1 -
       LEFT '                    Member Retention & Performance Analysis by
State and City (2016-2025)                                          ' SKIP 1 -
       LEFT
'+-------------------------------------------------------------------------
-------------------------------------------------------------+' SKIP 2 -
       LEFT 'STATE ANALYZED: ALL STATES' RIGHT 'PERIOD: 2016-2025' SKIP 1 -
       LEFT 'REPORT DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

--COLUMN FORMATTING
COLUMN state              HEADING 'STATE'             FORMAT A20
COLUMN city               HEADING 'CITY'              FORMAT A25
COLUMN retained_count     HEADING 'RETAINED|MEMBERS'  FORMAT 999,990
COLUMN lapsed_count       HEADING 'LAPSED|MEMBERS'    FORMAT 999,990
COLUMN retention_rate_pct HEADING 'RETENTION|RATE (%)'  FORMAT 990.9
COLUMN lapsed_rate_pct    HEADING 'LAPSED|RATE (%)'     FORMAT 990.9
COLUMN member_share_pct   HEADING 'STATE SHARE|(%)'     FORMAT 990.99

--REPORT BREAKS AND TOTALS
BREAK ON state SKIP 1 ON REPORT
```

```
COMPUTE SUM LABEL 'State Total:' OF retained_count lapsed_count ON state
COMPUTE AVG LABEL 'State Avg Retention Rate:' OF retention_rate_pct ON state
COMPUTE SUM LABEL 'Grand Total:' OF retained_count lapsed_count ON REPORT

--MAIN QUERY
WITH
-- Step A: Find the true first join date for every unique member
first_join AS (
  SELECT
    memberid, effective_start_date, memberState, memberCity
  FROM (
    SELECT
      memberid, effective_start_date, memberState, memberCity,
      ROW_NUMBER() OVER(PARTITION BY memberid ORDER BY effective_start_date
ASC) as rn
    FROM member_dim
  ) WHERE rn = 1
),
latest_status AS (
  SELECT
    memberid, is_current_flag
  FROM (
    SELECT
      memberid, is_current_flag,
      ROW_NUMBER() OVER(PARTITION BY memberid ORDER BY effective_start_date
DESC) as rn
    FROM member_dim
  ) WHERE rn = 1
),
member_cohort AS (
  SELECT
    memberid,
    memberState AS state,
    memberCity AS city
  FROM first_join
  WHERE EXTRACT(YEAR FROM effective_start_date) BETWEEN 2016 AND 2025
),
--Calculate comprehensive metrics for each city
city_metrics AS (
  SELECT
    mc.state,
    mc.city,
    COUNT(mc.memberid) AS total_members,
    COUNT(CASE WHEN ls.is_current_flag = '1' THEN 1 END) AS retained_count,
    COUNT(CASE WHEN ls.is_current_flag = '0' THEN 1 END) AS lapsed_count,
    SUM(COUNT(mc.memberid)) OVER (PARTITION BY mc.state) AS
state_total_members
  FROM member_cohort mc
  JOIN latest_status ls ON mc.memberid = ls.memberid
```

Sample Assignment Report

```
  GROUP BY mc.state, mc.city
)
--Assemble the final report with performance metrics
SELECT
  state,
  city,
  retained_count,
  lapsed_count,
  ROUND((retained_count / NULLIF(total_members, 0)) * 100, 1) AS
retention_rate_pct,
  ROUND((lapsed_count / NULLIF(total_members, 0)) * 100, 1) AS
lapsed_rate_pct,
  ROUND((total_members / state_total_members) * 100, 2) AS member_share_pct
FROM city_metrics
ORDER BY state, retention_rate_pct DESC, retained_count DESC;

--CLEANUP
TTITLE OFF
CLEAR BREAKS
CLEAR COLUMNS

SET HEADING ON
SET FEEDBACK ON

PROMPT
PROMPT
PROMPT ================================
PROMPT DRILL-DOWN: YEARLY TRENDS ANALYSIS
PROMPT ================================
PROMPT

SET DEFINE ON

ACCEPT p_state CHAR PROMPT 'Enter STATE name for yearly trend analysis: '

-- Setup for drill-down report
SET PAGESIZE 25
SET LINESIZE 80

TTITLE ON
TTITLE LEFT
'+-------------------------------------------------------------------------
--------------------------+' SKIP 1 -
       LEFT '            Yearly Member Trends for State: &p_state
(2016-2025)                        ' SKIP 1 -
       LEFT
'+-------------------------------------------------------------------------
--------------------------+' SKIP 2 -
       LEFT 'REPORT DATE: ' _DATE SKIP 2
```

Sample Assignment Report

```
-- Column formatting for drill-down
COLUMN join_year          HEADING 'YEAR'              FORMAT 9999
COLUMN new_members        HEADING 'NEW|MEMBERS'       FORMAT 999,990
COLUMN still_active       HEADING 'STILL|ACTIVE'      FORMAT 999,990
COLUMN have_lapsed        HEADING 'LAPSED|MEMBERS'    FORMAT 999,990
COLUMN yearly_retention   HEADING 'RETENTION|RATE (%)' FORMAT 990.9
COLUMN yearly_lapsed_rate HEADING 'LAPSED|RATE (%)'    FORMAT 990.9

BREAK ON REPORT
COMPUTE SUM LABEL 'Total:' OF new_members still_active have_lapsed ON REPORT

-- Drill-down query
WITH yearly_trends AS (
  SELECT
    EXTRACT(YEAR FROM fj.effective_start_date) AS join_year,
    COUNT(*) AS new_members,
    COUNT(CASE WHEN ls.is_current_flag = '1' THEN 1 END) AS still_active,
    COUNT(CASE WHEN ls.is_current_flag = '0' THEN 1 END) AS have_lapsed
  FROM (
    SELECT
      memberid, effective_start_date, memberState,
      ROW_NUMBER() OVER(PARTITION BY memberid ORDER BY effective_start_date
ASC) as rn
    FROM member_dim
    WHERE UPPER(memberState) = UPPER('&p_state')
  ) fj
  JOIN (
    SELECT
      memberid, is_current_flag,
      ROW_NUMBER() OVER(PARTITION BY memberid ORDER BY effective_start_date
DESC) as rn
    FROM member_dim
  ) ls ON fj.memberid = ls.memberid AND ls.rn = 1
  WHERE fj.rn = 1
    AND EXTRACT(YEAR FROM fj.effective_start_date) BETWEEN 2016 AND 2025
  GROUP BY EXTRACT(YEAR FROM fj.effective_start_date)
)
SELECT
  join_year,
  new_members,
  still_active,
  have_lapsed,
  ROUND((still_active / NULLIF(new_members, 0)) * 100, 1) AS
yearly_retention,
  ROUND((have_lapsed / NULLIF(new_members, 0)) * 100, 1) AS
yearly_lapsed_rate
FROM yearly_trends
ORDER BY join_year;
```

Sample Assignment Report

```
-- Final cleanup
TTITLE OFF
CLEAR BREAKS
CLEAR COLUMNS
UNDEFINE p_state
SET FEEDBACK ON
```

Sample Assignment Report

Output:

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES — PERIOD: 2016-2025
REPORT DATE: 08-SEP-25 — PAGE: 3

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
JOHOR	BEKOK	11	12	47.8	52.2	1.92
	AYER BALOI	11	12	47.8	52.2	1.92
	SIMPANG RENGAM	9	14	39.1	60.9	1.92
	ISKANDAR PUTERI	9	14	39.1	60.9	1.92
**********						
State Avg Retention				62.7		
State Total:		750	446			
KEDAH	KUALA KETIL	18	5	78.3	21.7	2.94
	MERBOK	18	5	78.3	21.7	2.94
	JENIANG	18	5	78.3	21.7	2.94
	KUALA LUMPURIM	18	5	78.3	21.7	2.94
	PADANG SERAI	18	5	78.3	21.7	2.94
	KUALA KEDAH	18	5	78.3	21.7	2.94
	ALOR SETAR	17	6	73.9	26.1	2.94
	GURUN	17	6	73.9	26.1	2.94
	KODIANG	16	7	69.6	30.4	2.94
	BANDAR BAHARU	16	7	69.6	30.4	2.94
	LANGGAR	16	7	69.6	30.4	2.94
	KOTA SARANG SEMUT	16	7	69.6	30.4	2.94
	SERDANG	16	7	69.6	30.4	2.94
	YAN	15	8	65.2	34.8	2.94
	SIK	15	8	65.2	34.8	2.94
	LANGKAWI	15	8	65.2	34.8	2.94

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES — PERIOD: 2016-2025
REPORT DATE: 08-SEP-25 — PAGE: 4

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
KEDAH	LUNAS	15	8	65.2	34.8	2.94
	KUALA PEGANG	15	8	65.2	34.8	2.94
	KUALA NERANG	14	9	60.9	39.1	2.94
	BALING	14	9	60.9	39.1	2.94
	KUPANG	14	9	60.9	39.1	2.94
	SIMPANG EMPAT	14	9	60.9	39.1	2.94
	PENDANG	14	9	60.9	39.1	2.94
	KOTA KUALA MUDA	14	9	60.9	39.1	2.94
	KARANGAN	13	10	56.5	43.5	2.94
	JITRA	13	10	56.5	43.5	2.94
	AYER HITAM	13	10	56.5	43.5	2.94
	UNIVERSITI UTARA MALAYSIA	13	10	56.5	43.5	2.94
	POKOK SENA	13	10	56.5	43.5	2.94
	BEDONG	12	11	52.2	47.8	2.94
	SUNGAI PETANI	12	11	52.2	47.8	2.94
	CHANGLOON	12	11	52.2	47.8	2.94
	BUKIT KAYU HITAM	10	13	43.5	56.5	2.94
	KEPALA BATAS	9	14	39.1	60.9	2.94
**********						
State Avg Retention				64.1		
State Total:		501	281			
KELANTAN	AYER LANAS	19	4	82.6	17.4	4.55
	PASIR MAS	18	5	78.3	21.7	4.55

Sample Assignment Report

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 5

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
KELANTAN	KETEREH	18	5	78.3	21.7	4.55
	KUALA KRAI	17	6	73.9	26.1	4.55
	TANAH MERAH	17	6	73.9	26.1	4.55
	GUA MUSANG	17	6	73.9	26.1	4.55
	KOTA BHARU	16	7	69.6	30.4	4.55
	TEMANGAN	16	7	69.6	30.4	4.55
	RANTAU PANJANG	16	7	69.6	30.4	4.55
	PASIR PUTEH	16	7	69.6	30.4	4.55
	MELOR	16	7	69.6	30.4	4.55
	DABONG	15	8	65.2	34.8	4.55
	TUMPAT	15	8	65.2	34.8	4.55
	CHERANG RUKU	15	8	65.2	34.8	4.55
	BACHOK	14	9	60.9	39.1	4.55
	MACHANG	14	9	60.9	39.1	4.55
	PULAI CHONDONG	14	9	60.9	39.1	4.55
	KUALA BALAH	14	9	60.9	39.1	4.55
	SELISING	14	9	60.9	39.1	4.55
	JELI	13	10	56.5	43.5	4.55
	KEM DESA PAHLAWAN	13	10	56.5	43.5	4.55
	WAKAF BHARU	9	14	39.1	60.9	4.55
**********						
State Avg Retention				66.4		
State Total:		336	170			

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 6

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
KUALA LUMPUR	KUALA LUMPUR	12	11	52.2	47.8	100.00
**********						
State Avg Retention				52.2		
State Total:		12	11			
LABUAN	LABUAN	18	5	78.3	21.7	100.00
**********						
State Avg Retention				78.3		
State Total:		18	5			
MELAKA	ALOR GAJAH	19	4	82.6	17.4	6.25
	SUNGAI UDANG	17	6	73.9	26.1	6.25
	TANJONG KLING	16	7	69.6	30.4	6.25
	ASAHAN	16	7	69.6	30.4	6.25
	SUNGAI RAMBAI	16	7	69.6	30.4	6.25
	MERLIMAU	16	7	69.6	30.4	6.25
	MELAKA	16	7	69.6	30.4	6.25
	AYER KEROH	15	8	65.2	34.8	6.25
	KEM TRENDAK	15	8	65.2	34.8	6.25
	LUBOK CHINA	15	8	65.2	34.8	6.25
	MASJID TANAH	14	9	60.9	39.1	6.25
	SELANDAR	13	10	56.5	43.5	6.25
	DURIAN TUNGGAL	13	10	56.5	43.5	6.25
	JASIN	13	10	56.5	43.5	6.25

Sample Assignment Report

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 7

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
MELAKA	KUALA SUNGAI BARU	12	11	52.2	47.8	6.25
	BEMBAN	11	12	47.8	52.2	6.25
**********						
State Avg Retention				64.4		
State Total:		237	131			
NEGERI SEMBILAN	LABU	18	5	78.3	21.7	3.85
	SIMPANG PERTANG	18	5	78.3	21.7	3.85
	BAHAU	17	6	73.9	26.1	3.85
	SIMPANG DURIAN	16	7	69.6	30.4	3.85
	GEMAS	16	7	69.6	30.4	3.85
	SEREMBAN	16	7	69.6	30.4	3.85
	RANTAU	16	7	69.6	30.4	3.85
	NIAI	16	7	69.6	30.4	3.85
	LINGGI	16	7	69.6	30.4	3.85
	KOTA	16	7	69.6	30.4	3.85
	SI RUSA	15	8	65.2	34.8	3.85
	KUALA PILAH	15	8	65.2	34.8	3.85
	KUALA KLAWANG	15	8	65.2	34.8	3.85
	NILAI	15	8	65.2	34.8	3.85
	PORT DICKSON	15	8	65.2	34.8	3.85
	MANTIN	14	9	60.9	39.1	3.85
	JOHOL	14	9	60.9	39.1	3.85
	BATU KIKIR	14	9	60.9	39.1	3.85

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 8

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
NEGERI SEMBILAN	TANJONG IPOH	14	9	60.9	39.1	3.85
	BANDAR ENSTEK	14	9	60.9	39.1	3.85
	BANDAR SERI JEMPOL	12	11	52.2	47.8	3.85
	REMBAU	11	12	47.8	52.2	3.85
	PUSAT BANDAR PALONG	11	12	47.8	52.2	3.85
	GEMENCHEH	11	12	47.8	52.2	3.85
	ROMPIN	11	12	47.8	52.2	3.85
	TAMPIN	10	13	43.5	56.5	3.85
**********						
State Avg Retention				62.9		
State Total:		376	222			
PAHANG	CHINI	19	4	82.6	17.4	2.63
	DAMAK	18	5	78.3	21.7	2.63
	MENTAKAB	18	5	78.3	21.7	2.63
	KEMAYAN	17	6	73.9	26.1	2.63
	GAMBANG	17	6	73.9	26.1	2.63
	CHENOR	17	6	73.9	26.1	2.63
	KUALA ROMPIN	17	6	73.9	26.1	2.63
	SUNGAI KOYAN	16	7	69.6	30.4	2.63
	KUALA KRAU	16	7	69.6	30.4	2.63
	BALOK	16	7	69.6	30.4	2.63
	TEMERLOH	16	7	69.6	30.4	2.63
	RAUB	16	7	69.6	30.4	2.63

Sample Assignment Report

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 9

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
PAHANG	LANCHANG	16	7	69.6	30.4	2.63
	BENTONG	16	7	69.6	30.4	2.63
	KUALA LIPIS	16	7	69.6	30.4	2.63
	KUANTAN	15	8	65.2	34.8	2.63
	PEKAN	15	8	65.2	34.8	2.63
	DONG	15	8	65.2	34.8	2.63
	JERANTUT	15	8	65.2	34.8	2.63
	SEGA	15	8	65.2	34.8	2.63
	GENTING HIGHLANDS	14	9	60.9	39.1	2.63
	BENTA	14	9	60.9	39.1	2.63
	BANDAR TUN ABDUL RAZAK	14	9	60.9	39.1	2.63
	BANDAR PUSAT JENGKA	14	9	60.9	39.1	2.63
	TANAH RATA	14	9	60.9	39.1	2.63
	PADANG TENGKU	14	9	60.9	39.1	2.63
	RINGLET	13	10	56.5	43.5	2.63
	KARAK	13	10	56.5	43.5	2.63
	LURAH BILUT	13	10	56.5	43.5	2.63
	BUKIT GOH	13	10	56.5	43.5	2.63
	BRINCHANG	13	10	56.5	43.5	2.63
	MUADZAM SHAH	13	10	56.5	43.5	2.63
	TRIANG	13	10	56.5	43.5	2.63
	BANDAR BERA	12	11	52.2	47.8	2.63
	MARAN	12	11	52.2	47.8	2.63
	BUKIT FRASER	12	11	52.2	47.8	2.63

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 10

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
PAHANG	BUKIT KUIN	12	11	52.2	47.8	2.63
	SUNGAI LEMBING	12	11	52.2	47.8	2.63
**********						
State Avg Retention				64.2		
State Total:		561	313			
PENANG	BALIK PULAU	18	5	78.3	21.7	5.00
	BAYAN LEPAS	17	6	73.9	26.1	5.00
	BATU MAUNG	17	6	73.9	26.1	5.00
	TANJUNG BUNGAH	15	8	65.2	34.8	5.00
	JELUTONG	15	8	65.2	34.8	5.00
	GELUGOR	15	8	65.2	34.8	5.00
	BUTTERWORTH	15	8	65.2	34.8	5.00
	SUNGAI JAWI	14	9	60.9	39.1	5.00
	TASEK GELUGOR	14	9	60.9	39.1	5.00
	PENANG HILL	14	9	60.9	39.1	5.00
	KUBANG SEMANG	14	9	60.9	39.1	5.00
	BATU FERRINGHI	14	9	60.9	39.1	5.00
	PERMATANG PAUH	14	9	60.9	39.1	5.00
	AYER ITAM	14	9	60.9	39.1	5.00
	PENAGA	13	10	56.5	43.5	5.00
	NIBONG TEBAL	13	10	56.5	43.5	5.00
	USM PULAU PINANG	13	10	56.5	43.5	5.00
	PERAI	12	11	52.2	47.8	5.00

Sample Assignment Report

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES — PERIOD: 2016-2025
REPORT DATE: 08-SEP-25 — PAGE: 11

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
PENANG	PULAU PINANG	12	11	52.2	47.8	5.00
	BUKIT MERTAJAM	12	11	52.2	47.8	5.00
**********						
State Avg Retention				62.0		
State Total:		285	175			
PERAK	BOTA	20	3	87.0	13.0	1.43
	LENGGONG	19	4	82.6	17.4	1.43
	TAIPING	18	5	78.3	21.7	1.43
	SUNGAI SUMUN	18	5	78.3	21.7	1.43
	SIMPANG	18	5	78.3	21.7	1.43
	KAMPAR	17	6	73.9	26.1	1.43
	KAMPUNG KEPAYANG	17	6	73.9	26.1	1.43
	MALIM NAWAR	17	6	73.9	26.1	1.43
	PADANG RENGAS	17	6	73.9	26.1	1.43
	PARIT BUNTAR	17	6	73.9	26.1	1.43
	SERI MANJUNG	17	6	73.9	26.1	1.43
	AYER TAWAR	17	6	73.9	26.1	1.43
	BIDOR	17	6	73.9	26.1	1.43
	CHEMOR	17	6	73.9	26.1	1.43
	GERIK	17	6	73.9	26.1	1.43
	JERAM	17	6	73.9	26.1	1.43
	BANDAR SERI ISKANDAR	16	7	69.6	30.4	1.43
	BEHRANG STESEN	16	7	69.6	30.4	1.43

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES — PERIOD: 2016-2025
REPORT DATE: 08-SEP-25 — PAGE: 12

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
PERAK	BRUAS	16	7	69.6	30.4	1.43
	CHENDERONG BALAI	16	7	69.6	30.4	1.43
	GOPENG	16	7	69.6	30.4	1.43
	KAMPUNG GAJAH	16	7	69.6	30.4	1.43
	SUNGAI SIPUT	16	7	69.6	30.4	1.43
	TELUK INTAN	16	7	69.6	30.4	1.43
	TEMOH	16	7	69.6	30.4	1.43
	MANONG	15	8	65.2	34.8	1.43
	TANJONG MALIM	15	8	65.2	34.8	1.43
	LANGKAP	15	8	65.2	34.8	1.43
	MATANG	15	8	65.2	34.8	1.43
	TLDM LUMUT	15	8	65.2	34.8	1.43
	SLIM RIVER	15	8	65.2	34.8	1.43
	BAGAN SERAI	15	8	65.2	34.8	1.43
	TAPAH ROAD	15	8	65.2	34.8	1.43
	ENGGOR	15	8	65.2	34.8	1.43
	ULU BERNAM	15	8	65.2	34.8	1.43
	INTAN	15	8	65.2	34.8	1.43
	IPOH	15	8	65.2	34.8	1.43
	KAMUNTING	15	8	65.2	34.8	1.43
	PARIT	15	8	65.2	34.8	1.43
	KUALA KURAU	15	8	65.2	34.8	1.43
	TANJONG TUALANG	15	8	65.2	34.8	1.43
	KUALA SEPETANG	14	9	60.9	39.1	1.43

Sample Assignment Report

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 13

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
PERAK	CHIKUS	14	9	60.9	39.1	1.43
	CHANGKAT JERING	14	9	60.9	39.1	1.43
	TROLAK	14	9	60.9	39.1	1.43
	LUMUT	14	9	60.9	39.1	1.43
	TANJONG RAMBUTAN	14	9	60.9	39.1	1.43
	TANJONG PIANDANG	14	9	60.9	39.1	1.43
	SITIAWAN	14	9	60.9	39.1	1.43
	SIMPANG AMPAT SEMANGGOL	14	9	60.9	39.1	1.43
	TRONG	14	9	60.9	39.1	1.43
	PENGKALAN HULU	14	9	60.9	39.1	1.43
	PANTAI REMIS	14	9	60.9	39.1	1.43
	BAGAN DATOH	13	10	56.5	43.5	1.43
	CHANGKAT KERUING	13	10	56.5	43.5	1.43
	SAUK	13	10	56.5	43.5	1.43
	HUTAN MELINTANG	13	10	56.5	43.5	1.43
	LAMBOR KANAN	13	10	56.5	43.5	1.43
	BATU GAJAH	13	10	56.5	43.5	1.43
	SUNGKAI	13	10	56.5	43.5	1.43
	PANGKOR	13	10	56.5	43.5	1.43
	TAPAH	13	10	56.5	43.5	1.43
	SELEKOH	12	11	52.2	47.8	1.43
	KUALA KANGSAR	12	11	52.2	47.8	1.43
	CHENDERIANG	12	11	52.2	47.8	1.43
	BATU KURAU	12	11	52.2	47.8	1.43

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 14

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
PERAK	PUSING	12	11	52.2	47.8	1.43
	ULU KINTA	11	12	47.8	52.2	1.43
	TRONOH	10	13	43.5	56.5	1.43
	SELAMA	9	14	39.1	60.9	1.43
**********						
State Avg Retention				64.5		
State Total:		1,039	571			
PERLIS	SIMPANG AMPAT	16	7	69.6	30.4	16.79
	KAKI BUKIT	15	8	65.2	34.8	16.79
	KUALA PERLIS	15	8	65.2	34.8	16.79
	PADANG BESAR	14	9	60.9	39.1	16.79
	KANGAR	12	10	54.5	45.5	16.06
	ARAU	10	13	43.5	56.5	16.79
**********						
State Avg Retention				59.8		
State Total:		82	55			
PUTRAJAYA	PUTRAJAYA	16	7	69.6	30.4	100.00
**********						
State Avg Retention				69.6		
State Total:		16	7			
SABAH	PAPAR	20	3	87.0	13.0	3.03

Sample Assignment Report

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES — PERIOD: 2016-2025
REPORT DATE: 08-SEP-25 — PAGE: 15

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
SABAH	PENAMPANG	20	3	87.0	13.0	3.03
	BEAUFORT	19	4	82.6	17.4	3.03
	KENINGAU	17	6	73.9	26.1	3.03
	NABAWAN	17	6	73.9	26.1	3.03
	KOTA KINABATANGAN	17	6	73.9	26.1	3.03
	PUTATAN	16	7	69.6	30.4	3.03
	SANDAKAN	16	7	69.6	30.4	3.03
	MENUMBOK	16	7	69.6	30.4	3.03
	LIKAS	16	7	69.6	30.4	3.03
	KUNAK	16	7	69.6	30.4	3.03
	BELURAN	16	7	69.6	30.4	3.03
	SIPITANG	15	8	65.2	34.8	3.03
	TAMBUNAN	15	8	65.2	34.8	3.03
	LAHAD DATU	14	9	60.9	39.1	3.03
	KOTA KINABALU	14	9	60.9	39.1	3.03
	INANAM	14	9	60.9	39.1	3.03
	BEVERLY	14	9	60.9	39.1	3.03
	TUARAN	14	9	60.9	39.1	3.03
	SEMPORNA	14	9	60.9	39.1	3.03
	BONGAWAN	13	10	56.5	43.5	3.03
	TENOM	13	10	56.5	43.5	3.03
	PAMOL	13	10	56.5	43.5	3.03
	KOTA BELUD	13	10	56.5	43.5	3.03
	KUALA PENYU	12	11	52.2	47.8	3.03

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES — PERIOD: 2016-2025
REPORT DATE: 08-SEP-25 — PAGE: 16

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
SABAH	KOTA MARUDU	12	11	52.2	47.8	3.03
	MEMBAKUT	12	11	52.2	47.8	3.03
	TAWAU	12	11	52.2	47.8	3.03
	TAMPARULI	12	11	52.2	47.8	3.03
	TENGHILAN	11	12	47.8	52.2	3.03
	TANJUNG ARU	10	13	43.5	56.5	3.03
	KUDAT	9	14	39.1	60.9	3.03
	RANAU	9	14	39.1	60.9	3.03
**********						
State Avg Retention				62.1		
State Total:		471	288			
SARAWAK	MATU	22	1	95.7	4.3	2.14
	BINTANGOR	19	4	82.6	17.4	2.14
	SUNDAR	18	4	81.8	18.2	2.04
	BELAWAI	18	5	78.3	21.7	2.14
	ROBAN	18	5	78.3	21.7	2.14
	SONG	18	5	78.3	21.7	2.14
	DEBAK	17	6	73.9	26.1	2.14
	SPAOH	17	6	73.9	26.1	2.14
	LIMBANG	16	6	72.7	27.3	2.04
	NANGA MEDAMIT	16	6	72.7	27.3	2.04
	BEKENU	16	7	69.6	30.4	2.14
	BETONG	16	7	69.6	30.4	2.14

Sample Assignment Report

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES — PERIOD: 2016-2025
REPORT DATE: 08-SEP-25 — PAGE: 17

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
SARAWAK	KOTA SAMARAHAN	16	7	69.6	30.4	2.14
	MUKAH	16	7	69.6	30.4	2.14
	JULAU	16	7	69.6	30.4	2.14
	KANOWIT	16	7	69.6	30.4	2.14
	BALINGIAN	16	7	69.6	30.4	2.14
	DALAT	15	8	65.2	34.8	2.14
	DARO	15	8	65.2	34.8	2.14
	SEBAUH	15	8	65.2	34.8	2.14
	SARIKEI	15	8	65.2	34.8	2.14
	LUNDU	15	8	65.2	34.8	2.14
	LUTONG	15	8	65.2	34.8	2.14
	SERIAN	15	8	65.2	34.8	2.14
	PUSA	14	9	60.9	39.1	2.14
	LUBOK ANTU	14	9	60.9	39.1	2.14
	LINGGA	14	9	60.9	39.1	2.14
	KUCHING	14	9	60.9	39.1	2.14
	KAPIT	14	9	60.9	39.1	2.14
	BINTULU	14	9	60.9	39.1	2.14
	LAWAS	13	9	59.1	40.9	2.04
	SRI AMAN	13	10	56.5	43.5	2.14
	SIMUNJAN	13	10	56.5	43.5	2.14
	ENGKILILI	13	10	56.5	43.5	2.14
	BELAGA	13	10	56.5	43.5	2.14
	BAU	12	11	52.2	47.8	2.14

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES — PERIOD: 2016-2025
REPORT DATE: 08-SEP-25 — PAGE: 18

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
SARAWAK	BARAM	12	11	52.2	47.8	2.14
	TATAU	12	11	52.2	47.8	2.14
	LONG LAMA	12	11	52.2	47.8	2.14
	NIAH	12	11	52.2	47.8	2.14
	MIRI	12	11	52.2	47.8	2.14
	SEBUYAU	12	11	52.2	47.8	2.14
	KABONG	10	13	43.5	56.5	2.14
	SARATOK	10	13	43.5	56.5	2.14
	SIBURAN	10	13	43.5	56.5	2.14
	ASAJAYA	10	13	43.5	56.5	2.14
	SIBU	10	13	43.5	56.5	2.14
**********						
State Avg Retention				63.1		
State Total:		679	398			
SELANGOR	PUCHONG	19	4	82.6	17.4	2.17
	BUKIT ROTAN	18	5	78.3	21.7	2.17
	SUBANG JAYA	18	5	78.3	21.7	2.17
	SUNGAI AYER TAWAR	18	5	78.3	21.7	2.17
	TANJONG KARANG	18	5	78.3	21.7	2.17
	RAWANG	18	5	78.3	21.7	2.17
	BATU CAVES	18	5	78.3	21.7	2.17
	AMPANG	17	6	73.9	26.1	2.17
	PULAU INDAH	17	6	73.9	26.1	2.17

Sample Assignment Report

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 19

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
SELANGOR	CYBERJAYA	17	6	73.9	26.1	2.17
	BANDAR PUNCAK ALAM	17	6	73.9	26.1	2.17
	SERENDAH	17	6	73.9	26.1	2.17
	SUNGAI BESAR	17	6	73.9	26.1	2.17
	SEKINCHAN	16	7	69.6	30.4	2.17
	SUNGAI PELEK	16	7	69.6	30.4	2.17
	SERI KEMBANGAN	16	7	69.6	30.4	2.17
	KAJANG	16	7	69.6	30.4	2.17
	TANJONG SEPAT	16	7	69.6	30.4	2.17
	SABAK BERNAM	15	8	65.2	34.8	2.17
	BANTING	15	8	65.2	34.8	2.17
	KLIA	15	8	65.2	34.8	2.17
	BERANANG	15	8	65.2	34.8	2.17
	PETALING JAYA	15	8	65.2	34.8	2.17
	SEMENYIH	15	8	65.2	34.8	2.17
	KUALA KUBU BARU	14	9	60.9	39.1	2.17
	JENJAROM	14	9	60.9	39.1	2.17
	BESTARI JAYA	14	9	60.9	39.1	2.17
	BANDAR BARU BANGI	14	9	60.9	39.1	2.17
	RASA	13	10	56.5	43.5	2.17
	TELOK PANGLIMA GARANG	13	10	56.5	43.5	2.17
	SUNGAI BULOH	13	10	56.5	43.5	2.17
	DENGKIL	13	10	56.5	43.5	2.17
	KAPAR	13	10	56.5	43.5	2.17

Member Retention & Performance Analysis by State and City (2016-2025)

STATE ANALYZED: ALL STATES
REPORT DATE: 08-SEP-25

PERIOD: 2016-2025
PAGE: 20

STATE	CITY	RETAINED MEMBERS	LAPSED MEMBERS	RETENTION RATE (%)	LAPSED RATE (%)	STATE SHARE (%)
SELANGOR	KERLING	13	10	56.5	43.5	2.17
	PULAU CAREY	13	10	56.5	43.5	2.17
	KUALA SELANGOR	12	11	52.2	47.8	2.17
	CHERAS	12	11	52.2	47.8	2.17
	SHAH ALAM	12	11	52.2	47.8	2.17
	HULU LANGAT	12	11	52.2	47.8	2.17
	SEPANG	12	11	52.2	47.8	2.17
	PULAU KETAM	11	12	47.8	52.2	2.17
	BATANG KALI	11	12	47.8	52.2	2.17
	KLANG	11	12	47.8	52.2	2.17
	PELABUHAN KLANG	11	12	47.8	52.2	2.17
	KUANG	10	13	43.5	56.5	2.17
	BATU ARANG	9	14	39.1	60.9	2.17
**********						
State Avg Retention				63.2		
State Total:		669	389			
TERENGGANU	DUNGUN	18	5	78.3	21.7	4.55
	PAKA	18	5	78.3	21.7	4.55
	MARANG	17	6	73.9	26.1	4.55
	JERTEH	16	7	69.6	30.4	4.55
	PERMAISURI	16	7	69.6	30.4	4.55
	SUNGAI TONG	16	7	69.6	30.4	4.55
	CENEH	16	7	69.6	30.4	4.55

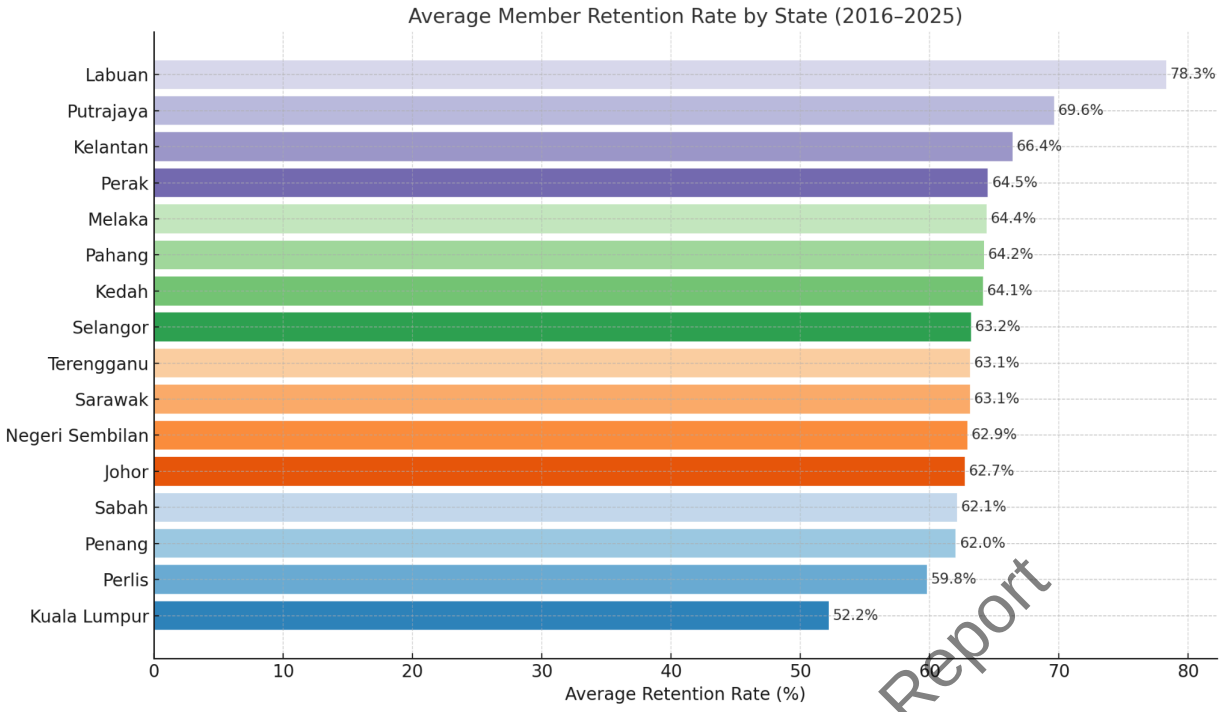
Sample Assignment Report

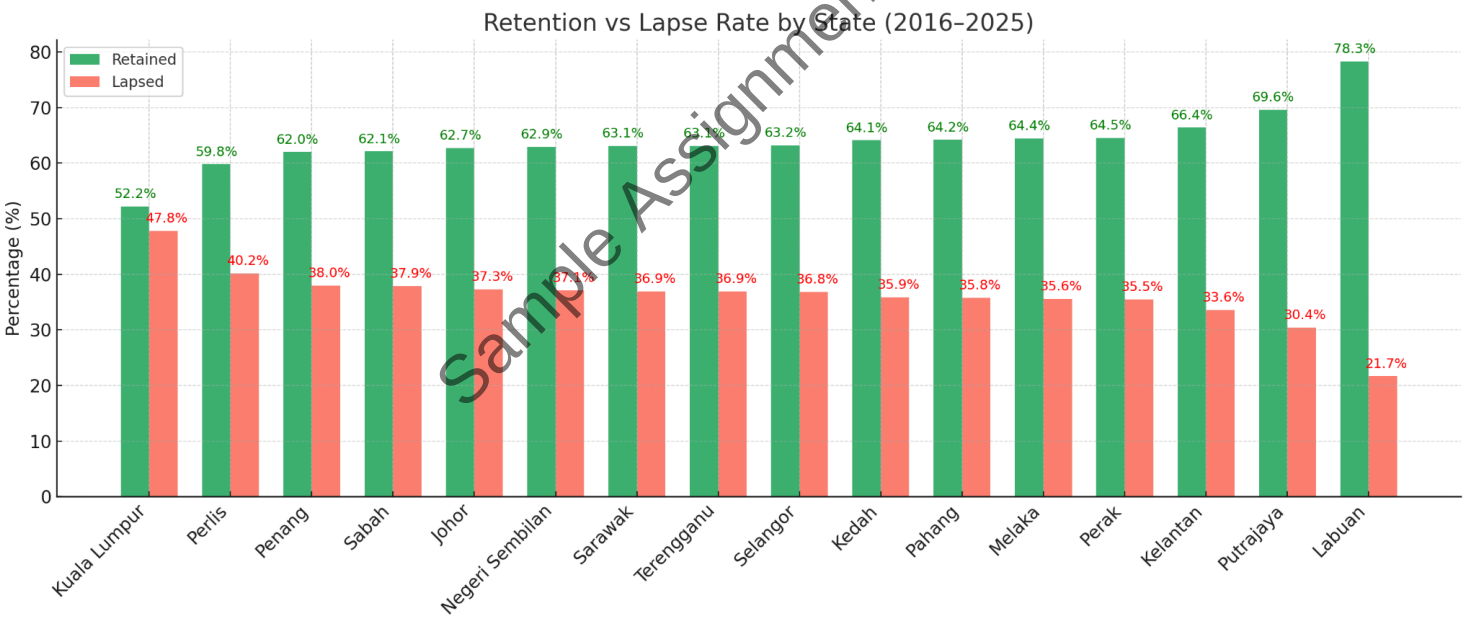
```
+------------------------------------------------------------------------------------------------------------------------------
                 Member Retention & Performance Analysis by State and City (2016-2025)
+------------------------------------------------------------------------------------------------------------------------------

STATE ANALYZED: ALL STATES                                                                                   PERIOD: 2016-2025
REPORT DATE: 08-SEP-25                                                                                              PAGE:   21

                                                      RETAINED   LAPSED RETENTION   LAPSED STATE SHARE
STATE                CITY                              MEMBERS  MEMBERS  RATE (%) RATE (%)         (%)
-------------------- ------------------------------- -------- -------- --------- -------- -----------
TERENGGANU           KUALA TERENGGANU                      16        7      69.6     30.4        4.55
                     CUKAI                                 15        8      65.2     34.8        4.55
                     CHALOK                                15        8      65.2     34.8        4.55
                     KUALA BESUT                           15        8      65.2     34.8        4.55
                     KEMASEK                               14        9      60.9     39.1        4.55
                     KERTEH                                14        9      60.9     39.1        4.55
                     KUALA BERANG                          14        9      60.9     39.1        4.55
                     KIJAL                                 14        9      60.9     39.1        4.55
                     BUKIT PAYONG                          14        9      60.9     39.1        4.55
                     KETENGAH JAYA                         13       10      56.5     43.5        4.55
                     BUKIT BESI                            13       10      56.5     43.5        4.55
                     KAMPUNG RAJA                          13       10      56.5     43.5        4.55
                     AJIL                                  11       12      47.8     52.2        4.55
                     AYER PUTEH                            11       12      47.8     52.2        4.55
                     AL MUKTATFI BILLAH SHAH               10       13      43.5     56.5        4.55
********************                                 -------- -------- ---------
State Avg Retention                                                       63.1
State Total:                                              319      187

                                                     -------- --------
Grand Total:                                            6,351    3,649
```

```
==================================
DRILL-DOWN: YEARLY TRENDS ANALYSIS
==================================

Enter STATE name for yearly trend analysis: selangor

+------------------------------------------------------------------------------------------
            Yearly Member Trends for State: selangor (2016-2025)
+------------------------------------------------------------------------------------------

REPORT DATE: 08-SEP-25

           NEW    STILL   LAPSED RETENTION   LAPSED
 YEAR  MEMBERS   ACTIVE  MEMBERS  RATE (%) RATE (%)
----- -------- -------- -------- --------- --------
 2018       54       45        9      83.3     16.7
 2019       42       27       15      64.3     35.7
 2020       42       32       10      76.2     23.8
 2021      194      155       39      79.9     20.1
 2022      236      160       76      67.8     32.2
 2023      234      132      102      56.4     43.6
 2024      256      118      138      46.1     53.9
      -------- -------- --------
Total    1,058      669      389
```

Sample Assignment Report

Visualisation:





Sample Assignment Report

Justification:

This report analyses member acquisition and retention by state and city from 2016 to 2025 to show where members remain active and where they lapse, supporting targeted renewal actions. The results reveal that state-level retention is highest in Labuan with 78.3 percent, followed by Kelantan at 66.4 percent, Perak at 64.5 percent, Selangor at 63.2 percent, Sarawak at 63.1 percent, and Terengganu also at 63.1 percent. In contrast, Kuala Lumpur has the lowest retention at 52.2 percent. At the city level, Kahang in Johor demonstrates strong retention at 82.6 percent with 19 retained out of 23 members, while Simpang Rengam in Johor shows a lower rate at 39.1 percent. The yearly drill-down of Selangor shows how retention has changed over time, with 83.3 percent in 2018 dropping to 46.1 percent in 2024. These variations matter because they signal areas of strength or weakness in sustaining library membership. Higher retention means members continue to engage with the library, while lower retention suggests disengagement and potential loss of the customer base.

Each metric is calculated in a clear step-by-step method. City retention rate is calculated by dividing the number of retained members in a city by the total number of members in that city and then multiplying by one hundred. City lapsed rate is calculated by dividing the number of lapsed members by the total number of members in that city and multiplying by one hundred. Yearly cohort retention is found by dividing the number of still-active members from a join year by the number of new members who joined in that year and then multiplying by one hundred. Similarly, the yearly cohort lapse rate is calculated by dividing the number of lapsed members from a join year by the number of new members from that year and multiplying by one hundred.

Sample Assignment Report

Each column in the report is included for a specific reason. State and City columns identify the geographical area being analyzed. Retained Members show the number of individuals who remain active, while Lapsed Members capture those who have dropped out. Retention Rate and Lapsed Rate provide fair comparisons between different-sized locations. State Average Retention gives a benchmark to evaluate city-level results. State Total shows how many members are involved in each state overall. Grand Total rolls up the national-level results. In the yearly drill-down, the Year column identifies the join cohort, New Members shows the total who joined that year, Still Active Members reflect those who are still engaged, and Lapsed Members reflect those who are not. The yearly cohort rates then summarise how well each cohort has been retained over time.

This data is important because it shows what is happening with membership trends, why certain areas are performing better or worse, and where targeted action is needed. The changes occurred due to differences in member behavior across locations and years, such as variations in service delivery, communication, or life stage transitions. Knowing who has lapsed and where allows the library to take informed actions to retain more members and prevent membership decline.

In the future, the library should identify cities where the retention rate is lower than the average retention rate of their respective state. For example, if Selangor's state average retention rate is 63.2%, but Shah Alam shows a retention rate of only 47.5%, then Shah Alam should be prioritized. These cities should then be ranked by their Lapsed Rate in descending order, followed by total number of lapsed members to focus resources where the drop-off is greatest. For each city on the list, the library staff should generate member renewal lists by selecting members whose latest status is marked as "lapsed." These members should receive a fixed sequence of renewal reminders such as three SMS messages over two weeks to their registered phone numbers, and the response outcomes should be tracked at the city level to assess effectiveness.To further improve renewal rates, the library

can organize “Renewal Day” pop-up events in these low-retention cities. For instance, if Simpang Rengam in Johor has a low retention rate of 39.1%, the library could set up a booth at a local mall or community center, invite lapsed members from the city, and process renewals on-site. After the event, the conversion rate should be calculated by comparing how many of the invited lapsed members reactivated their accounts and comparing the city’s retention rate before and after the event.To evaluate long-term campaign effectiveness, the library should match member join dates with specific campaign periods, then calculate the acquisition lift and twelve-month retention rate for that cohort. For example, if 500 members joined during the January 2023 campaign and 320 remained active in January 2024, the 12-month retention rate would be 64%, which can be compared against non-campaign joiners. These actions will enable the library to optimize retention efforts, increase renewal success, and sustain a stable and engaged member base across all regions.

Sample Assignment Report

3.1.3 Supplier Performance and Cost Analysis

SQL Code:

```
SET PAGESIZE 35
SET LINESIZE 130
SET VERIFY OFF
SET FEEDBACK OFF
SET DEFINE OFF

--REPORT TITLE
TTITLE ON
TTITLE LEFT
'+-------------------------------------------------------------------------
-------------------------------------------------------+' SKIP 1 -
       LEFT '                 Member Retention & Performance Analysis by
State and City (2016-2025)                                          ' SKIP 1 -
       LEFT
'+-------------------------------------------------------------------------
-------------------------------------------------------+' SKIP 2 -
       LEFT 'STATE ANALYZED: ALL STATES' RIGHT 'PERIOD: 2016-2025' SKIP 1 -
       LEFT 'REPORT DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

--COLUMN FORMATTING
COLUMN state              HEADING 'STATE'               FORMAT A20
COLUMN city               HEADING 'CITY'                FORMAT A25
COLUMN retained_count     HEADING 'RETAINED|MEMBERS'  FORMAT 999,990
COLUMN lapsed_count       HEADING 'LAPSED|MEMBERS'    FORMAT 999,990
COLUMN retention_rate_pct HEADING 'RETENTION|RATE (%)'  FORMAT 990.9
COLUMN lapsed_rate_pct    HEADING 'LAPSED|RATE (%)'     FORMAT 990.9
COLUMN member_share_pct   HEADING 'STATE SHARE|(%)'     FORMAT 990.99

--REPORT BREAKS AND TOTALS
BREAK ON state SKIP 1 ON REPORT
COMPUTE SUM LABEL 'State Total:' OF retained_count lapsed_count ON state
COMPUTE AVG LABEL 'State Avg Retention Rate:' OF retention_rate_pct ON state
COMPUTE SUM LABEL 'Grand Total:' OF retained_count lapsed_count ON REPORT

--MAIN QUERY
WITH
-- Step A: Find the true first join date for every unique member
first_join AS (
  SELECT
    memberid, effective_start_date, memberState, memberCity
  FROM (
    SELECT
      memberid, effective_start_date, memberState, memberCity,
      ROW_NUMBER() OVER(PARTITION BY memberid ORDER BY effective_start_date
ASC) as rn
```

Sample Assignment Report

```
    FROM member_dim
  ) WHERE rn = 1
),
latest_status AS (
  SELECT
    memberid, is_current_flag
  FROM (
    SELECT
      memberid, is_current_flag,
      ROW_NUMBER() OVER(PARTITION BY memberid ORDER BY effective_start_date
DESC) as rn
    FROM member_dim
  ) WHERE rn = 1
),
member_cohort AS (
  SELECT
    memberid,
    memberState AS state,
    memberCity AS city
  FROM first_join
  WHERE EXTRACT(YEAR FROM effective_start_date) BETWEEN 2016 AND 2025
),
--Calculate comprehensive metrics for each city
city_metrics AS (
  SELECT
    mc.state,
    mc.city,
    COUNT(mc.memberid) AS total_members,
    COUNT(CASE WHEN ls.is_current_flag = '1' THEN 1 END) AS retained_count,
    COUNT(CASE WHEN ls.is_current_flag = '0' THEN 1 END) AS lapsed_count,
    SUM(COUNT(mc.memberid)) OVER (PARTITION BY mc.state) AS
state_total_members
  FROM member_cohort mc
  JOIN latest_status ls ON mc.memberid = ls.memberid
  GROUP BY mc.state, mc.city
)
--Assemble the final report with performance metrics
SELECT
  state,
  city,
  retained_count,
  lapsed_count,
  ROUND((retained_count / NULLIF(total_members, 0)) * 100, 1) AS
retention_rate_pct,
  ROUND((lapsed_count / NULLIF(total_members, 0)) * 100, 1) AS
lapsed_rate_pct,
  ROUND((total_members / state_total_members) * 100, 2) AS member_share_pct
FROM city_metrics
ORDER BY state, retention_rate_pct DESC, retained_count DESC;
```

Sample Assignment Report

```
--CLEANUP
TTITLE OFF
CLEAR BREAKS
CLEAR COLUMNS

SET HEADING ON
SET FEEDBACK ON

PROMPT
PROMPT
PROMPT ==================================
PROMPT DRILL-DOWN: YEARLY TRENDS ANALYSIS
PROMPT ==================================
PROMPT

SET DEFINE ON

ACCEPT p_state CHAR PROMPT 'Enter STATE name for yearly trend analysis: '

-- Setup for drill-down report
SET PAGESIZE 25
SET LINESIZE 80

TTITLE ON
TTITLE LEFT
'+-------------------------------------------------------------------------
-------------------------+' SKIP 1 -
       LEFT '            Yearly Member Trends for State: &p_state
(2016-2025)                    ' SKIP 1 -
       LEFT
'+-------------------------------------------------------------------------
-------------------------+' SKIP 2 -
       LEFT 'REPORT DATE: ' _DATE SKIP 2

-- Column formatting for drill-down
COLUMN join_year          HEADING 'YEAR'              FORMAT 9999
COLUMN new_members        HEADING 'NEW|MEMBERS'       FORMAT 999,990
COLUMN still_active       HEADING 'STILL|ACTIVE'      FORMAT 999,990
COLUMN have_lapsed        HEADING 'LAPSED|MEMBERS'    FORMAT 999,990
COLUMN yearly_retention   HEADING 'RETENTION|RATE (%)' FORMAT 990.9
COLUMN yearly_lapsed_rate HEADING 'LAPSED|RATE (%)'    FORMAT 990.9

BREAK ON REPORT
COMPUTE SUM LABEL 'Total:' OF new_members still_active have_lapsed ON REPORT

-- Drill-down query
WITH yearly_trends AS (
  SELECT
```

Sample Assignment Report

```
    EXTRACT(YEAR FROM fj.effective_start_date) AS join_year,
    COUNT(*) AS new_members,
    COUNT(CASE WHEN ls.is_current_flag = '1' THEN 1 END) AS still_active,
    COUNT(CASE WHEN ls.is_current_flag = '0' THEN 1 END) AS have_lapsed
  FROM (
    SELECT
      memberid, effective_start_date, memberState,
      ROW_NUMBER() OVER(PARTITION BY memberid ORDER BY effective_start_date
ASC) as rn
    FROM member_dim
    WHERE UPPER(memberState) = UPPER('&p_state')
  ) fj
  JOIN (
    SELECT
      memberid, is_current_flag,
      ROW_NUMBER() OVER(PARTITION BY memberid ORDER BY effective_start_date
DESC) as rn
    FROM member_dim
  ) ls ON fj.memberid = ls.memberid AND ls.rn = 1
  WHERE fj.rn = 1
    AND EXTRACT(YEAR FROM fj.effective_start_date) BETWEEN 2016 AND 2025
  GROUP BY EXTRACT(YEAR FROM fj.effective_start_date)
)
SELECT
  join_year,
  new_members,
  still_active,
  have_lapsed,
  ROUND((still_active / NULLIF(new_members, 0)) * 100, 1) AS
yearly_retention,
  ROUND((have_lapsed / NULLIF(new_members, 0)) * 100, 1) AS
yearly_lapsed_rate
FROM yearly_trends
ORDER BY join_year;

-- Final cleanup
TTITLE OFF
CLEAR BREAKS
CLEAR COLUMNS
UNDEFINE p_state
SET FEEDBACK ON
```

Sample Assignment Report

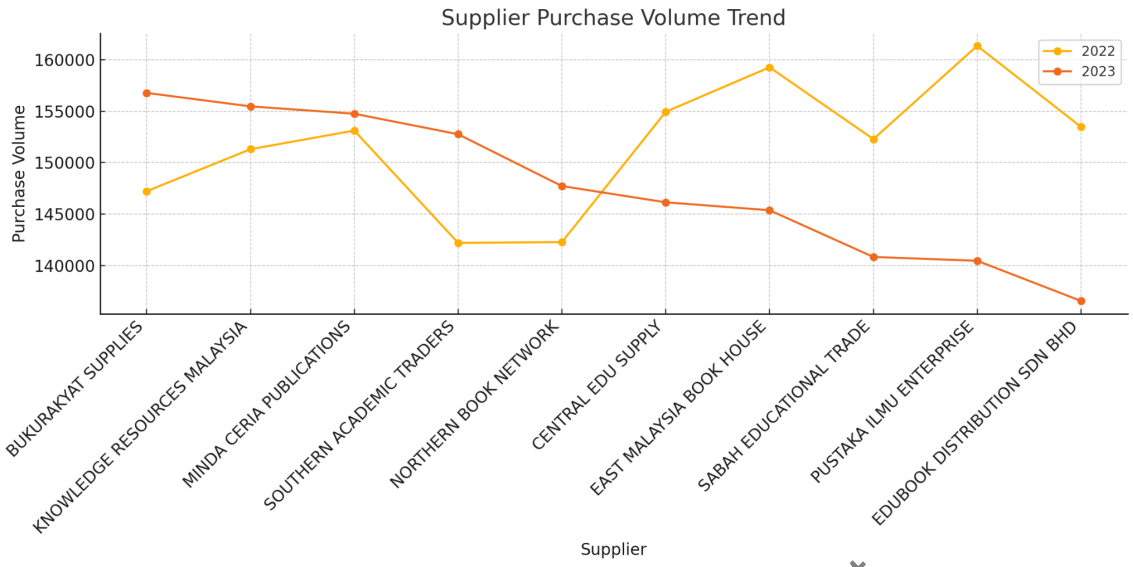
Output:

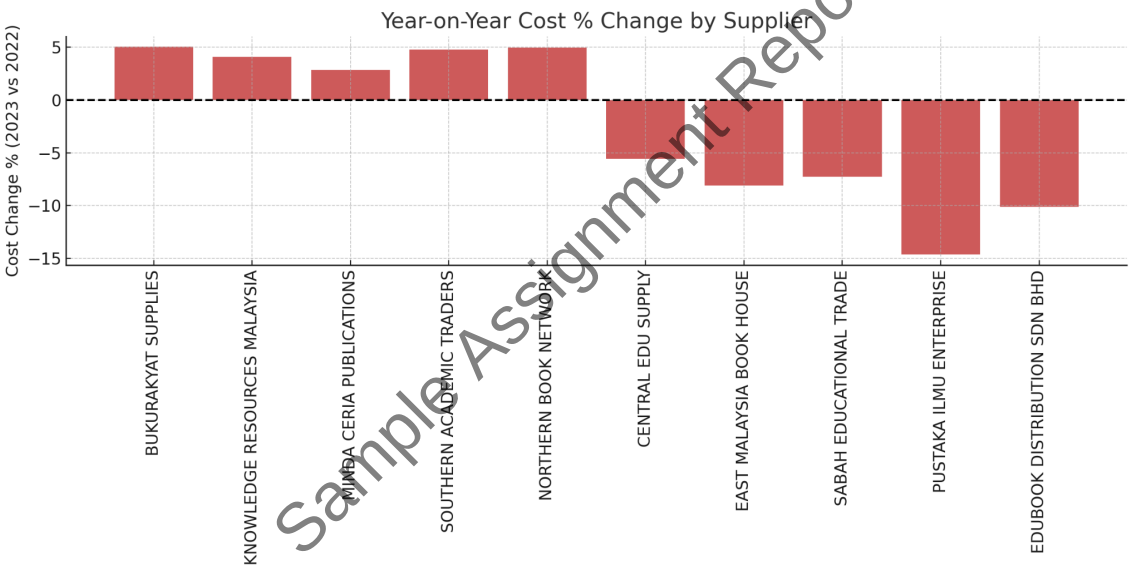
```
SQL> @ C:\DW assignment\q5.txt
Enter the CURRENT year (e.g., 2024): 2023
Enter the PREVIOUS year (e.g., 2023): 2022

 ----------------------------------------------------------------------------------------------------------------------------------
                                                SUPPLIER PERFORMANCE AND COST ANALYSIS
 ----------------------------------------------------------------------------------------------------------------------------------

DATE: 08-SEP-25
PERIOD:2022-2023                                                                                                          PAGE:    1

                                                          Vol          2023          2022              2023       2022
      RANK                              2023        2022  Grwth         Total         Total            AvgUnit    AvgUnit
      2023 SUPPLIER                   PurVol      PurVol     %          Cost          Cost   Cost%       Cost       Cost
---------- -------------------- ------------ ------------ -------- -------------- -------------- ------- ---------- ----------
         1 BUKURAKYAT SUPPLIES       156,766      147,233     6.47   9,747,618.54   9,278,573.54    5.06      62.63      62.69
         2 KNOWLEDGE RESOURCES       155,459      151,308     2.74   9,758,704.59   9,375,028.33    4.09      63.22      62.02
           MALAYSIA

         3 MINDA CERIA PUBLICAT      154,746      153,115     1.07   9,738,330.28   9,467,280.19    2.86      62.53      61.97
           IONS

         4 SOUTHERN ACADEMIC TR      152,759      142,222     7.41   9,310,446.73   8,884,120.20    4.80      60.76      62.66
           ADERS

         5 NORTHERN BOOK NETWOR      147,732      142,301     3.82   9,319,028.54   8,876,710.51    4.98      62.99      62.72
           K

         6 CENTRAL EDU SUPPLY        146,157      154,935    -5.67   9,151,707.82   9,693,408.07   -5.59      62.02      62.69
         7 EAST MALAYSIA BOOK H      145,389      159,251    -8.70   8,967,332.09   9,757,938.94   -8.10      61.72      61.43
           OUSE

         8 SABAH EDUCATIONAL TR      140,858      152,284    -7.50   8,830,612.87   9,521,894.03   -7.26      63.07      62.89
           ADE

         9 PUSTAKA ILMU ENTERPR      140,490      161,330   -12.92   8,715,690.95  10,212,637.91  -14.66      61.81      63.25
           ISE
```

```
 ----------------------------------------------------------------------------------------------------------------------------------
                                                SUPPLIER PERFORMANCE AND COST ANALYSIS
 ----------------------------------------------------------------------------------------------------------------------------------

DATE: 08-SEP-25
PERIOD:2022-2023                                                                                                          PAGE:    2

                                                          Vol          2023          2022              2023       2022
      RANK                              2023        2022  Grwth         Total         Total            AvgUnit    AvgUnit
      2023 SUPPLIER                   PurVol      PurVol     %          Cost          Cost   Cost%       Cost       Cost
---------- -------------------- ------------ ------------ -------- -------------- -------------- ------- ---------- ----------
        10 EDUBOOK DISTRIBUTION      136,598      153,472   -10.99   8,549,639.31   9,511,240.99  -10.11      62.65      62.03
            SDN BHD

                                ------------ ------------          -------------- --------------         ---------- ----------
AVG:                                                                                                          62.34      62.43
TOTAL:                             1,476,954    1,517,451           92,089,111.72  94,578,832.71

10 rows selected.
```

Sample Assignment Report

Visualisation:





Sample Assignment Report

Justification:

This report analyses supplier performance and cost trends between 2022 and 2023 to identify high-volume partners and detect opportunities for cost savings. The dataset shows a moderate decline in overall purchase volume and total spend, with the average unit cost holding steady across suppliers. The top three suppliers in 2023 by purchase volume are Buyurakyat Supplies (156,766 units), Knowledge Resources Malaysia (155,459 units), and Minda Ceria Publications (154,746 units). These suppliers also demonstrate positive year-over-year growth: Buyurakyat (+6.47%), Knowledge Resources Malaysia (+2.74%), and Minda Ceria (+1.07%). In contrast, Pustaka Ilmu Enterprise shows the steepest decline at –12.92%, followed by EduBook Distribution Sdn Bhd at –10.99%. From a cost perspective, most suppliers that experienced volume drops also delivered cost reductions, with Pustaka Ilmu recording the largest drop in total spend (–14.66%) and Central EDU Supply achieving a 5.59% reduction in cost.

Conversely, suppliers like Buyurakyat Supplies and Northern Book Network not only increased volume but also saw an increase in total spending, indicating possible price inflation.

Each column in this report plays a role in decision-making. Rank orders suppliers by 2023 purchase volume, highlighting the largest contributors to the library's supply chain. 2023 PurVol and 2022 PurVol show the quantity of units purchased per year, enabling direct volume comparison. Vol Growth rate quantifies year-over-year growth by measuring the change in volume as a percentage of the 2022 baseline. 2023 Total Cost and 2022 Total Cost capture total annual expenditure per supplier. Cost rate expresses the change in total cost from 2022 to 2023. 2023 AvgUnit Cost and 2022 AvgUnit Cost calculate average unit prices to offer a basis for pricing comparison. The AVG row summarizes mean unit costs across all suppliers, while the TOTAL row indicates the overall purchase volume and spend for each year.

The core formulas used in this report are as follows. Vol Growth rate is calculated by subtracting the 2022 purchase volume from the 2023 purchase volume, dividing the result by the 2022 volume, and multiplying by 100. Cost rate is derived by subtracting the 2022 total cost from the 2023 total cost, dividing by the 2022 cost, and multiplying by 100. Average Unit Cost for a year is calculated by dividing the total cost by the purchase volume for that year. Rank is assigned by sorting all suppliers in descending order of 2023 purchase volume.

In the future, library teams should designate strategic partners by identifying suppliers who deliver the highest volume of books while maintaining low or stable average unit costs, as these provide the best value. For example, Southern Academic Traders showed strong performance with a 7.41% increase in purchase volume and a relatively low average unit cost of RM60.76, making them a suitable candidate for bulk orders in future cycles. For top suppliers with high purchase volume but rising unit prices, such as Knowledge Resources Malaysia whose unit cost increased from RM62.02 to RM63.22, staff should request a targeted price review to bring rates back in line with market benchmarks. When total spending rises faster than volume, the library should negotiate price reductions to prevent budget overspending. If a supplier's purchase volume drops while prices remain high, such as Pustaka Ilmu Enterprise with a −12.92% drop in volume and a high unit cost of RM63.25, new orders should be limited, and demand redirected to better-performing alternatives. Library staff should schedule negotiations with specific unit cost targets and deadlines, revise future purchase allocations based on supplier performance, and maintain cost-efficiency while ensuring consistent book supply.

Sample Assignment Report

3.2 Goh Ee Lin

3.2.1 Annual Customer Value (ACV) & Segmentation Analysis

SQL Code:

```
CLEAR BREAKS
CLEAR COMPUTES
CLEAR COLUMNS
SET VERIFY OFF
SET FEEDBACK OFF
SET PAGESIZE 50
SET LINESIZE 80
-- A4 Paper Size Formatting

TTITLE ON
TTITLE CENTER '=========================================================' SKIP 1
-
       CENTER 'Annual Customer Value (ACV) and Segmentation Analysis' SKIP 1
-
       CENTER '=========================================================' SKIP 1
-
       LEFT 'Report Generated on: ' _DATE -
       RIGHT 'Page: ' SQL.PNO SKIP 2

COLUMN tier FORMAT A15 HEADING 'Tier'
COLUMN year FORMAT 9999 HEADING 'Year'
COLUMN pct_total_members FORMAT A20 HEADING '% of Total Members'
COLUMN avg_acv FORMAT 999,990.99 HEADING 'Avg ACV (RM)'
COLUMN pct_tier_at_risk FORMAT A20 HEADING '% of Tier at Risk'

BREAK ON tier SKIP 1

-- compute the average ACV for all members within a tier over the selected
years
COMPUTE AVG LABEL 'Tier Avg ACV:' OF avg_acv ON tier

-- View 1: Aggregates annual sales and fines for each member by year
CREATE OR REPLACE VIEW MEMBER_ANNUAL_VALUE_V AS
SELECT
    COALESCE(s.member_key, l.member_key) AS member_key,
    COALESCE(s.cal_year, l.cal_year) AS cal_year,
    NVL(s.total_sales, 0) + NVL(l.total_fines, 0) AS annual_acv,
    NVL(s.total_sales, 0) AS annual_sales,
    NVL(l.total_fines, 0) AS annual_fines
FROM
    (
        -- Aggregate total sales per member per year
        SELECT
```

Sample Assignment Report

```
            sf.member_key,
            dd.cal_year,
            SUM(sf.orderTotalPrice) AS total_sales
        FROM SALES_FACT sf
        JOIN DATE_DIM dd ON sf.date_key = dd.date_key
        GROUP BY sf.member_key, dd.cal_year
    ) s
FULL OUTER JOIN
    (
        -- Aggregate total fines per member per year
        SELECT
            lf.member_key,
            dd.cal_year,
            SUM(lf.totalFine) AS total_fines
        FROM LOAN_FACT lf
        JOIN DATE_DIM dd ON lf.date_key = dd.date_key
        WHERE lf.totalFine > 0
        GROUP BY lf.member_key, dd.cal_year
    ) l ON s.member_key = l.member_key AND s.cal_year = l.cal_year;

-- View 2: Determines annual tier and risk status year-over-year for each
member
CREATE OR REPLACE VIEW MEMBER_ANNUAL_TIERS_V AS
SELECT
    mav.member_key,
    mav.cal_year,
    mav.annual_acv,
    mav.annual_sales,
    mav.annual_fines,
    -- Check if current year's value is less than the previous year's
    CASE
        WHEN mav.annual_acv < LAG(mav.annual_acv, 1, 0) OVER (PARTITION BY
mav.member_key ORDER BY mav.cal_year)
        THEN 1
        ELSE 0
    END AS is_at_risk,
    -- Assign tiers based on the ANNUAL value
    CASE
        WHEN mav.annual_acv >= 750 THEN 'Platinum'
        WHEN mav.annual_acv >= 350 THEN 'Gold'
        WHEN mav.annual_acv >= 100 THEN 'Silver'
        ELSE 'Bronze'
    END AS tier,
    -- Previous year's ACV for trend analysis
    LAG(mav.annual_acv, 1, 0) OVER (PARTITION BY mav.member_key ORDER BY
mav.cal_year) AS prev_year_acv
FROM MEMBER_ANNUAL_VALUE_V mav;

-- View 3: Combines tier information with member demographics for age group
```

Sample Assignment Report

```
analysis
CREATE OR REPLACE VIEW MEMBER_DEMOGRAPHIC_TIERS_V AS
SELECT
    mat.member_key,
    mat.cal_year,
    mat.annual_acv,
    mat.annual_sales,
    mat.annual_fines,
    mat.is_at_risk,
    mat.tier,
    mat.prev_year_acv,
    md.memberName,
    md.memberEmail,
    md.dob,
    md.memberCountry,
    md.memberState,
    md.memberCity,
    -- Calculate age group based on the analysis year
    CASE
        WHEN TRUNC((TO_DATE(mat.cal_year || '-12-31', 'YYYY-MM-DD') -
md.dob) / 365.25) <= 25 THEN '18-25'
        WHEN TRUNC((TO_DATE(mat.cal_year || '-12-31', 'YYYY-MM-DD') -
md.dob) / 365.25) <= 35 THEN '26-35'
        WHEN TRUNC((TO_DATE(mat.cal_year || '-12-31', 'YYYY-MM-DD') -
md.dob) / 365.25) <= 45 THEN '36-45'
        WHEN TRUNC((TO_DATE(mat.cal_year || '-12-31', 'YYYY-MM-DD') -
md.dob) / 365.25) > 45 THEN '46+'
        ELSE 'Unknown'
    END AS age_group,
    -- Calculate exact age for the analysis year
    TRUNC((TO_DATE(mat.cal_year || '-12-31', 'YYYY-MM-DD') - md.dob) /
365.25) AS age_in_year
FROM MEMBER_ANNUAL_TIERS_V mat
JOIN MEMBER_DIM md ON mat.member_key = md.member_key
WHERE md.is_current_flag = '1';

PROMPT
PROMPT Please provide the analysis period for the ACV Report.
ACCEPT start_year_prompt CHAR PROMPT 'Enter Start Year (e.g., 2016): '
ACCEPT end_year_prompt CHAR PROMPT 'Enter End Year (e.g., 2025): '
PROMPT

-- Report Section 1: ACV and Customer Segmentation Report
WITH
-- Aggregate the results by tier and year for the final report.
ANNUAL_TIER_SUMMARY AS (
    SELECT
        tier,
        cal_year,
```

Sample Assignment Report

```
        COUNT(DISTINCT member_key) AS member_count,
        AVG(annual_acv) AS avg_acv,
        SUM(is_at_risk) AS at_risk_count
    FROM MEMBER_ANNUAL_TIERS_V
    WHERE cal_year BETWEEN TO_NUMBER('&start_year_prompt') AND
TO_NUMBER('&end_year_prompt')
    GROUP BY tier, cal_year
),
-- Step 4: Calculate final percentages using window functions for totals.
FINAL_REPORT AS (
    SELECT
        ats.tier,
        ats.cal_year AS year,
        ats.member_count,
        ats.avg_acv,
        ats.at_risk_count,
        -- Use a window function to get the total number of members for each
year to calculate percentages.
        SUM(ats.member_count) OVER (PARTITION BY ats.cal_year) AS
total_members_in_year
    FROM ANNUAL_TIER_SUMMARY ats
)
-- Final presentation query to format the output as requested.
SELECT
    fr.tier,
    fr.year,
    TO_CHAR(ROUND((fr.member_count * 100.0 / fr.total_members_in_year), 2),
'990.99MI') || '%' AS pct_total_members,
    ROUND(fr.avg_acv, 2) AS avg_acv,
    TO_CHAR(ROUND((fr.at_risk_count * 100.0 / fr.member_count), 2),
'990.99MI') || '%' AS pct_tier_at_risk
FROM FINAL_REPORT fr
ORDER BY
    -- Custom order for tiers
    CASE fr.tier
        WHEN 'Platinum' THEN 1
        WHEN 'Gold' THEN 2
        WHEN 'Silver' THEN 3
        WHEN 'Bronze' THEN 4
    END,
    fr.year;

CLEAR COMPUTES
CLEAR BREAKS
CLEAR COLUMNS

-- Report Section 2: Drill-Down Report by Age Group
ACCEPT drilldown_year_prompt CHAR PROMPT 'Enter Year to Drill Down (e.g.,
2023): '
```

Sample Assignment Report

```
ACCEPT drilldown_tier_prompt CHAR PROMPT 'Enter Tier to Drill Down (e.g.,
Gold): '

TTITLE CENTER '=========================================================' SKIP
1 -
       CENTER 'Drill-Down Analysis for &drilldown_tier_prompt Tier in
&drilldown_year_prompt by Age Group' SKIP 1 -
       CENTER '=========================================================' SKIP
2

COLUMN age_group FORMAT A10 HEADING 'Age Group'
COLUMN member_count_drilldown FORMAT 999,999 HEADING 'Member Count'
COLUMN pct_of_tier FORMAT A20 HEADING '% of Tier Members'
COLUMN avg_acv_age_group FORMAT '999,990.99' HEADING 'Avg ACV (RM)'
COLUMN pct_at_risk_age_group FORMAT A15 HEADING '% At Risk'

-- Use the demographic view for age group analysis
WITH
DRILLDOWN_SUMMARY AS (
    SELECT
        age_group,
        COUNT(*) AS member_count_drilldown,
        AVG(annual_acv) AS avg_acv_age_group,
        SUM(is_at_risk) AS at_risk_count,
        SUM(COUNT(*)) OVER () as total_tier_members
    FROM MEMBER_DEMOGRAPHIC_TIERS_V
    WHERE cal_year = TO_NUMBER('&drilldown_year_prompt')
      AND UPPER(tier) = UPPER('&drilldown_tier_prompt')
    GROUP BY age_group
)
SELECT
    age_group,
    member_count_drilldown,
    TO_CHAR(ROUND((member_count_drilldown * 100.0 / total_tier_members), 2),
'990.99MI') || '%' AS pct_of_tier,
    avg_acv_age_group,
    CASE WHEN member_count_drilldown > 0 THEN TO_CHAR(ROUND((at_risk_count *
100.0 / member_count_drilldown), 2), '990.99MI') || '%' ELSE '0.00%' END AS
pct_at_risk_age_group
FROM DRILLDOWN_SUMMARY
ORDER BY age_group;


CLEAR COLUMNS
UNDEFINE start_year_prompt
UNDEFINE end_year_prompt
UNDEFINE drilldown_year_prompt
UNDEFINE drilldown_tier_prompt
CLEAR BREAKS
```

Sample Assignment Report

```
CLEAR COMPUTES
SET FEEDBACK ON
TTITLE OFF
PROMPT Report complete.
PROMPT
```

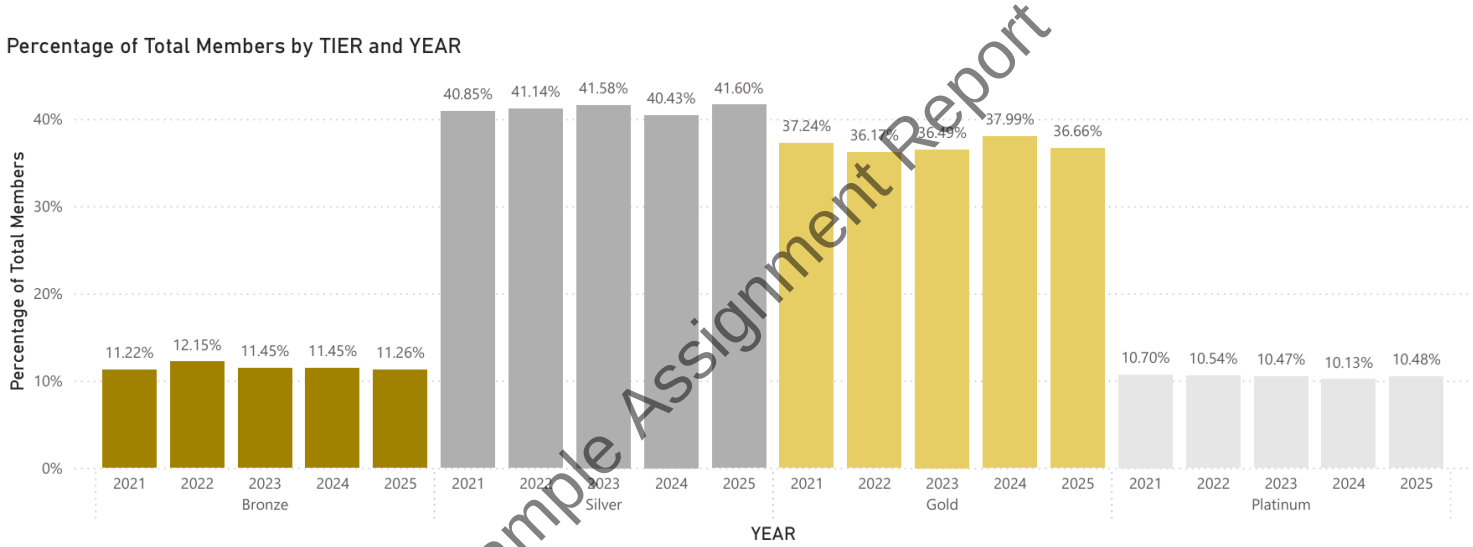
Output:

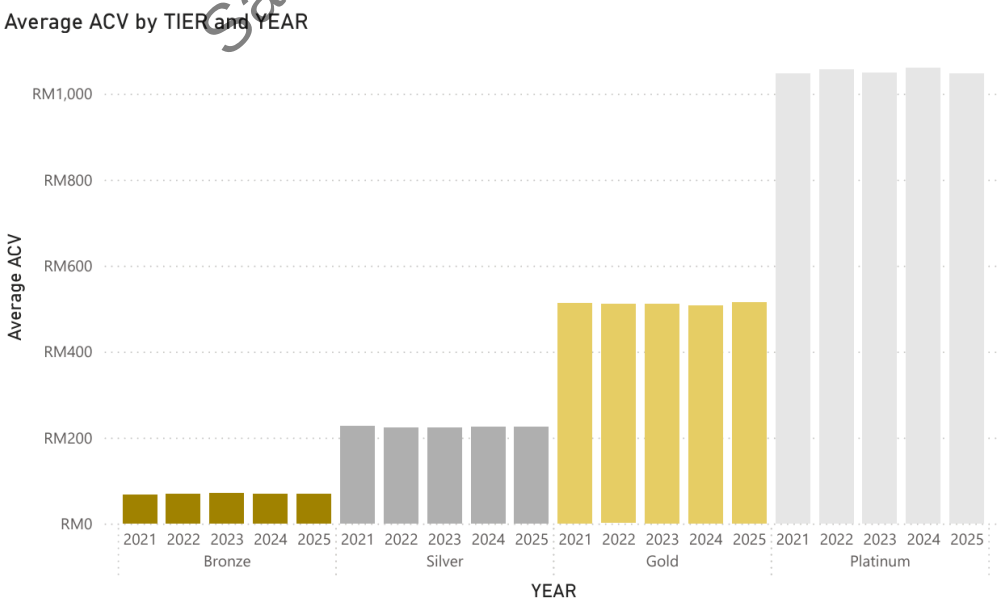
```
Please provide the analysis period for the ACV Report.
Enter Start Year (e.g., 2016): 2021
Enter End Year (e.g., 2025): 2025


              ====================================================
              Annual Customer Value (ACV) and Segmentation Analysis
              ====================================================
Report Generated on: 08-SEP-25                                 Page:        1

Tier            Year % of Total Members   Avg ACV (RM) % of Tier at Risk
--------------- ----- -------------------- ------------ --------------
Platinum        2021  10.70 %                 1,047.36   6.76 %
                2022  10.54 %                 1,056.83   7.93 %
                2023  10.47 %                 1,048.19   9.45 %
                2024  10.13 %                 1,060.12   9.73 %
                2025  10.48 %                 1,046.24   8.07 %
***************                            ------------
Tier Avg ACV:                                 1,051.75

Gold            2021  37.24 %                   512.59  26.97 %
                2022  36.17 %                   509.68  29.26 %
                2023  36.49 %                   510.98  28.25 %
                2024  37.99 %                   507.22  27.01 %
                2025  36.66 %                   514.15  28.64 %
***************                            ------------
Tier Avg ACV:                                   510.92

Silver          2021  40.85 %                   225.89  67.17 %
                2022  41.14 %                   223.59  67.98 %
                2023  41.58 %                   222.73  67.31 %
                2024  40.43 %                   224.15  67.05 %
                2025  41.60 %                   224.17  67.38 %
***************                            ------------
Tier Avg ACV:                                   224.11

Bronze          2021  11.22 %                    67.60  93.31 %
                2022  12.15 %                    68.33  95.26 %
                2023  11.45 %                    69.84  93.52 %
                2024  11.45 %                    67.92  95.17 %
                2025  11.26 %                    68.28  94.30 %
***************                            ------------
Tier Avg ACV:                                    68.39
```

Sample Assignment Report

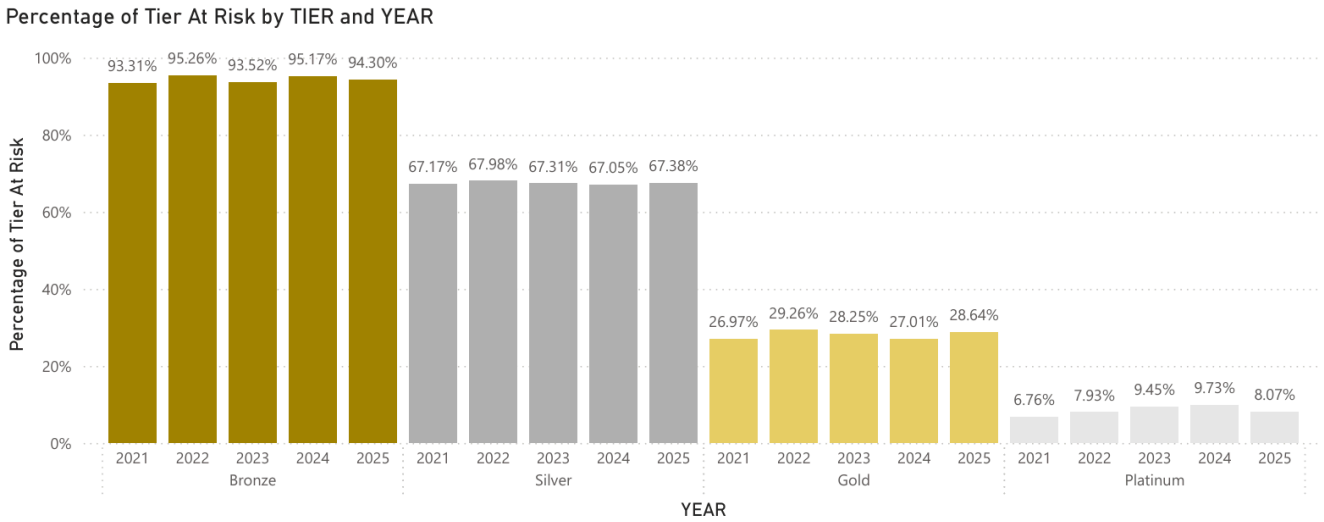
```
Enter Year to Drill Down (e.g., 2023): 2025
Enter Tier to Drill Down (e.g., Gold): Gold

              ====================================================
              Drill-Down Analysis for Gold Tier in 2025 by Age Group
              ====================================================

Age Group  Member Count % of Tier Members    Avg ACV (RM) % At Risk
---------- ------------ -------------------- ------------ ---------------
18-25               155   5.77 %                   521.16  33.55 %
26-35               547  20.37 %                   515.44  28.52 %
36-45               547  20.37 %                   505.83  27.06 %
46+               1,436  53.48 %                   516.07  28.76 %
Report complete.
```

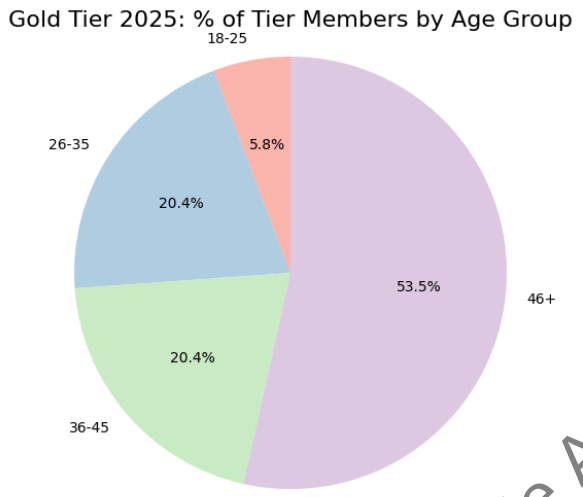
Visualisation:

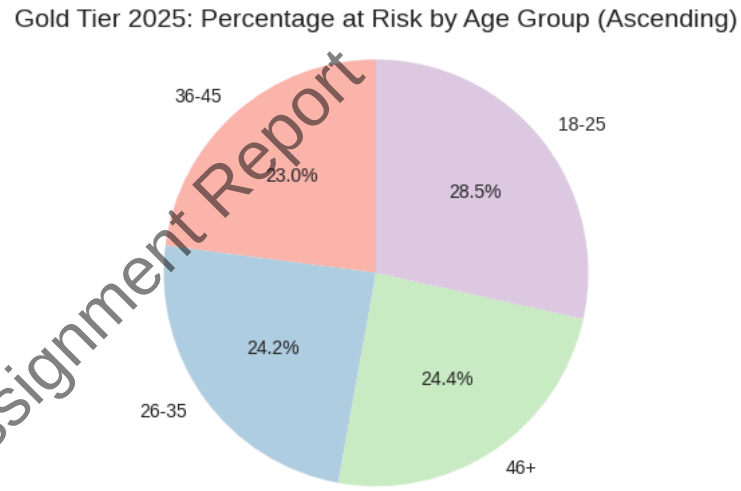




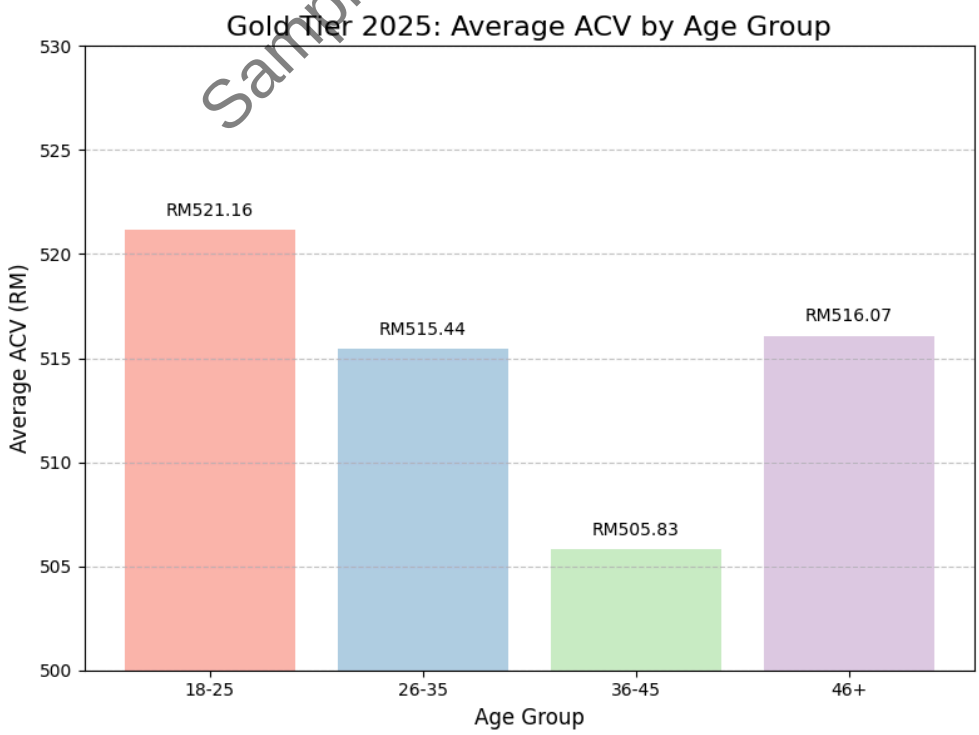
Sample Assignment Report







Sample Assignment Report



Justification:

This report shows a clear picture of the library's member base, segmented by value (Annual Customer Value - ACV) and risk of churn. Member's ACV is calculated by adding their total annual purchases from the library to their total annual overdue fines. A member is classified as 'at risk' if their ACV for the current year is lower than their ACV from the previous year, indicating a decline in their engagement and value. The query uses fixed monetary thresholds to segment its members into tiers annually. Members with an Annual Customer Value (ACV) of RM 750 or more are classified as Platinum. Those with an ACV between RM 350 and RM 749.99 qualify for the Gold tier. Members with an ACV between RM 100 and RM 349.99 are placed in the Silver tier, while those with an ACV of less than RM 100 fall into the Bronze tier.

The majority of members are in the Silver (~41%) and Gold (~37%) tiers. This distribution has remained remarkably consistent over the last five years. The highest-value (Platinum) and lowest-value (Bronze) members make up smaller, roughly equal portions of the base (~10-12% each). High-value (Platinum) members are the most loyal, with a low at-risk percentage. On the other hand, lower tiers (Bronze and Silver) show a massive percentage of "at-risk" members (93-95% and ~67% respectively). While these members have low individual value, they make up nearly 80% of the total member base. Their disengagement is a leading indicator of future membership cancellations, which could shrink the overall member pool and reduce the number of customers who could potentially be upgraded to higher-value tiers.

The 2025 drill-down on the Gold tier reveals that while the '46+' age group forms the backbone of the tier (53.48% of its members), the youngest '18-25' group has the highest percentage of at-risk members (33.55%) which is a threat to the library's long-term health as this demographic represents the next generation of loyal, high-value customers. If they are already disengaging, the pipeline to replenish the Platinum tier in the future is weak. Members aged 18-25 are often in transitional life stages (students, new professionals). Their library usage might be intense for a short period (e.g., buying academic texts for a semester) and then drop off sharply once that need is met. As digital natives, they are also more likely to switch to more convenient digital or online alternatives.

To address these problems, in the future, library directors should consider iImplement an automated system that identifies members who have not made a purchase or loan in the past six months and sends them personalized re-engagement emails. These emails should highlight new arrivals in the categories they have previously purchased or borrowed, making the content directly relevant to their interests. For example, if a member has a history of borrowing business and finance books, the email could feature the latest titles in that category, along with a short review or staff recommendation. They can also develop a basic loyalty program where members can earn points for purchases and loans, creating an incentive for repeat visits. Then, using the points they earned, they can redeem some benefits, such as extending loan duration in the future. The program should be tiered to encourage members to move up the value ladder — for example, Gold members could earn points at a slightly higher rate than Silver members, creating an incentive to increase their activity. Additionally, for retaining young Gold members, library directors should consider developing marketing campaigns on platforms they actively use (Instagram, TikTok, YouTube) and promoting popular, high-demand categories such as fantasy, romance, and mystery. The library should also ensure ample stock of trending bestsellers from authors popular with this demographic, reducing the risk of them turning to online competitors. Engagement could be boosted through interactive elements like polls ("Vote for the next book we stock"), reading challenges, or short-form video

Sample Assignment Report

reviews by librarians.

3.2.2 Geographic Customer Hotspot Analysis

SQL Code:

```
SET VERIFY OFF
SET FEEDBACK OFF
SET PAGESIZE 50
SET LINESIZE 80

ACCEPT start_year_prompt CHAR PROMPT 'Enter Start Year for Analysis (e.g.,
2016): '
ACCEPT end_year_prompt CHAR PROMPT 'Enter End Year for Analysis (e.g.,
2025): '
PROMPT

PROMPT Creating temporary view for analysis...
CREATE OR REPLACE VIEW GEO_CITY_PERFORMANCE_V AS
SELECT
    md.memberState,
    md.memberCity,
    SUM(sf.orderTotalPrice) AS sales_in_geo,
    COUNT(DISTINCT sf.member_key) AS members_in_geo
FROM SALES_FACT sf
JOIN DATE_DIM dd ON sf.date_key = dd.date_key
JOIN MEMBER_DIM md ON sf.member_key = md.member_key
WHERE
    dd.cal_year BETWEEN TO_NUMBER('&start_year_prompt') AND
TO_NUMBER('&end_year_prompt')
GROUP BY
    md.memberState,
    md.memberCity;

PROMPT View created. Generating reports...
PROMPT

-- Report Section 1: State-Level Summary
TTITLE CENTER '=====================================================' SKIP 1
-
       CENTER 'State-Level Geographic Performance Summary' SKIP 1 -
       CENTER 'For Years &start_year_prompt to &end_year_prompt' SKIP 1 -
       CENTER '=====================================================' SKIP
1-
       LEFT 'Report Generated on: ' _DATE  -
       RIGHT 'Page: ' SQL.PNO SKIP 2

-- Define column formats
COLUMN member_state FORMAT A20 HEADING 'State'
```

Sample Assignment Report

```
COLUMN avg_sales_per_member FORMAT '999,999.99' HEADING 'Avg Sales (RM)|Per
Member'
COLUMN pct_active_members FORMAT A15 HEADING '% of Active|Members'
COLUMN pct_total_revenue FORMAT A15 HEADING '% of Total|Revenue'

-- The query aggregates the city-level data from the view to the state
level.
WITH
STATE_AGGREGATES AS (
    SELECT
        memberState as member_state,
        SUM(sales_in_geo) as sales_in_state,
        SUM(members_in_geo) as members_in_state
    FROM GEO_CITY_PERFORMANCE_V
    GROUP BY memberState
),
PERIOD_TOTALS AS (
    SELECT
        SUM(sales_in_geo) AS total_sales_in_period,
        SUM(members_in_geo) AS total_members_in_period
    FROM GEO_CITY_PERFORMANCE_V
)
SELECT
    sa.member_state,
    sa.sales_in_state / sa.members_in_state AS avg_sales_per_member,
    TO_CHAR(ROUND((sa.members_in_state * 100.0 /
pt.total_members_in_period), 2), '990.99') || '%' AS pct_active_members,
    TO_CHAR(ROUND((sa.sales_in_state * 100.0 / pt.total_sales_in_period),
2), '990.99') || '%' AS pct_total_revenue
FROM
    STATE_AGGREGATES sa,
    PERIOD_TOTALS pt
WHERE
    sa.members_in_state > 0
ORDER BY
    avg_sales_per_member DESC;


-- Report Section 2: Interactive City-Level Drill-Down
PROMPT
PROMPT
ACCEPT drilldown_state_prompt CHAR PROMPT 'Enter State to see city-level
detail (or press Enter to skip): '
ACCEPT top_n_prompt CHAR PROMPT 'Enter the number of Top Cities to display
(e.g., 5): '
PROMPT

CLEAR COLUMNS
TTITLE CENTER '=====================================================' SKIP 1
```

Sample Assignment Report

```
-
       CENTER 'Top &top_n_prompt City-Level Detail for
&drilldown_state_prompt' SKIP 1 -
       CENTER 'For Years &start_year_prompt to &end_year_prompt' SKIP 1 -
       CENTER '=====================================================' SKIP 2

COLUMN ranking FORMAT 999 HEADING 'Rnk'
COLUMN member_city FORMAT A20 HEADING 'City'
COLUMN members_in_geo FORMAT 999,999 HEADING 'Members'
COLUMN avg_sales_per_member FORMAT '999,999.99' HEADING 'Avg Sales'
COLUMN pct_active_members FORMAT A10 HEADING '% Members'
COLUMN pct_total_revenue FORMAT A10 HEADING '% Revenue'
COLUMN cumulative_pct_revenue FORMAT A10 HEADING 'Cumulative|% Rev'

-- This query runs ONLY if a state was entered. It now filters for the Top N
cities.
WITH
STATE_TOTALS AS (
    SELECT
        SUM(sales_in_geo) as total_sales_in_state,
        SUM(members_in_geo) as total_members_in_state
    FROM GEO_CITY_PERFORMANCE_V
    WHERE UPPER(memberState) = UPPER('&drilldown_state_prompt')
),
RANKED_CITIES AS (
    SELECT
        RANK() OVER (ORDER BY (ca.sales_in_geo / ca.members_in_geo) DESC) as
ranking,
        ca.memberCity as member_city,
        ca.members_in_geo,
        ca.sales_in_geo,
        ca.sales_in_geo / ca.members_in_geo AS avg_sales_per_member,
        (ca.members_in_geo * 100.0 / st.total_members_in_state) AS
pct_active_members,
        (ca.sales_in_geo * 100.0 / st.total_sales_in_state) AS
pct_total_revenue
    FROM
        GEO_CITY_PERFORMANCE_V ca,
        STATE_TOTALS st
    WHERE
        UPPER(ca.memberState) = UPPER('&drilldown_state_prompt')
        AND ca.members_in_geo > 0
),
CUMULATIVE_CITIES AS (
    SELECT
        rc.*,
        SUM(pct_total_revenue) OVER (ORDER BY ranking) as
cumulative_pct_revenue
    FROM RANKED_CITIES rc
```

Sample Assignment Report

```
)
SELECT
    ranking,
    member_city,
    members_in_geo,
    avg_sales_per_member,
    TO_CHAR(ROUND(pct_active_members, 1), '990.9') || '%' AS
pct_active_members,
    TO_CHAR(ROUND(pct_total_revenue, 1), '990.9') || '%' AS
pct_total_revenue,
    TO_CHAR(ROUND(cumulative_pct_revenue, 1), '990.9') || '%' AS
cumulative_pct_revenue
FROM CUMULATIVE_CITIES
WHERE ranking <= TO_NUMBER('&top_n_prompt')
ORDER BY
    ranking;


PROMPT
PROMPT Dropping temporary view...
DROP VIEW GEO_CITY_PERFORMANCE_V;
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
UNDEFINE start_year_prompt
UNDEFINE end_year_prompt
UNDEFINE drilldown_state_prompt
UNDEFINE top_n_prompt
SET FEEDBACK ON
SET VERIFY ON
TTITLE OFF
PROMPT
PROMPT Report complete.
PROMPT
```

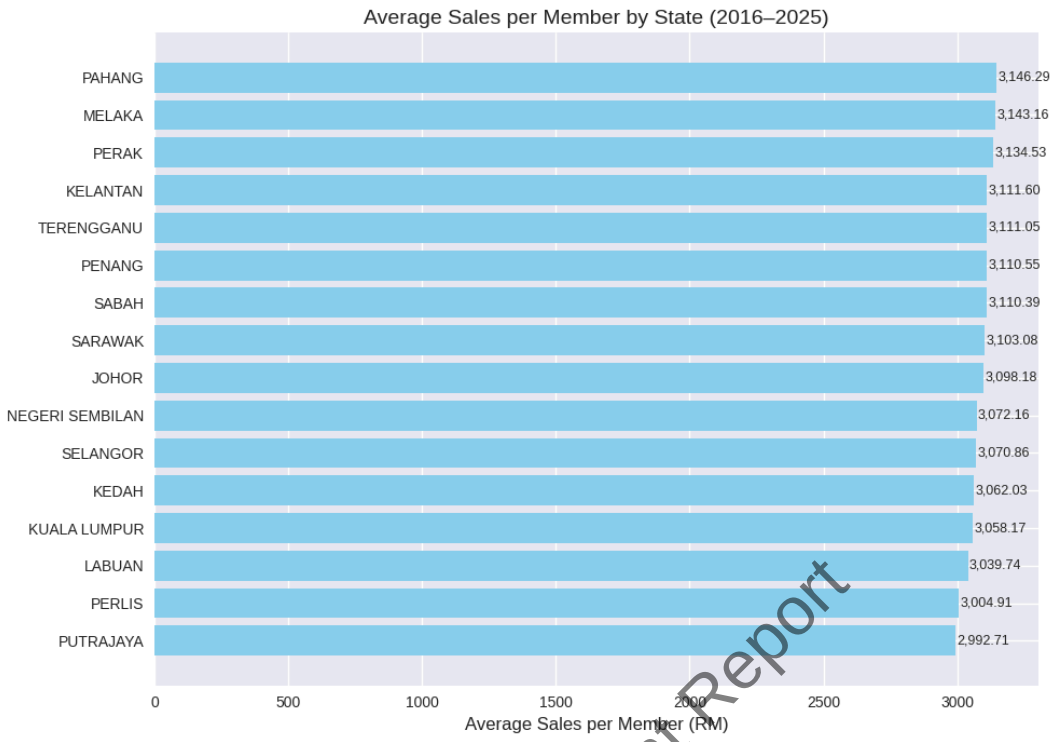
Sample Assignment Report

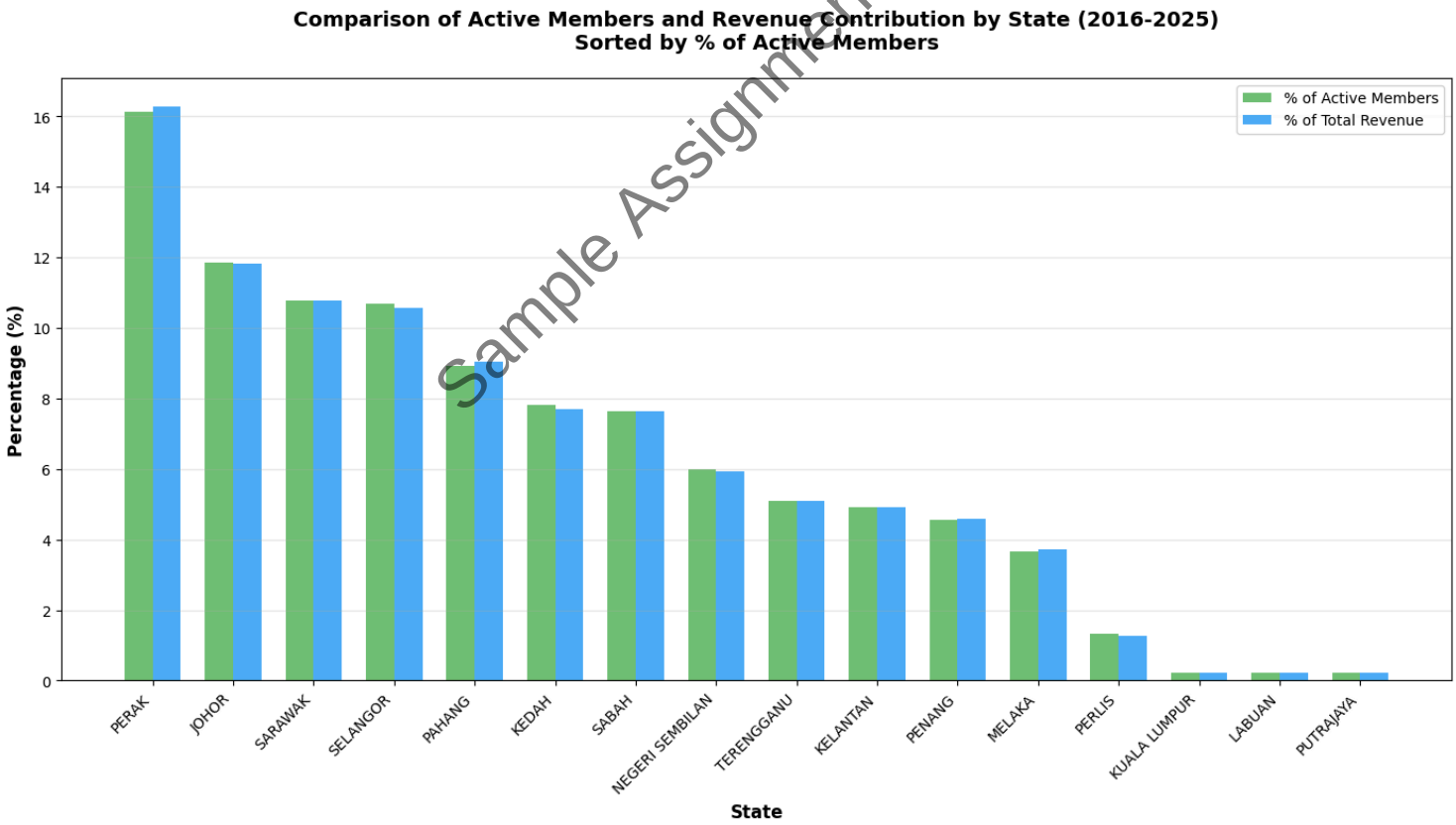
Output:

```
Enter Start Year for Analysis (e.g., 2016): 2016
Enter End Year for Analysis (e.g., 2025): 2025

Creating temporary view for analysis...
View created. Generating reports...


              ====================================================
                   State-Level Geographic Performance Summary
                              For Years 2016 to 2025
              ====================================================
Report Generated on: 08-SEP-25                                 Page:          1

                     Avg Sales (RM) % of Active     % of Total
State                    Per Member Members         Revenue
-------------------- -------------- --------------- ---------------
PAHANG                     3,146.29    8.92%           9.04%
MELAKA                     3,143.16    3.66%           3.71%
PERAK                      3,134.53   16.12%          16.27%
KELANTAN                   3,111.60    4.92%           4.93%
TERENGGANU                 3,111.05    5.09%           5.10%
PENANG                     3,110.55    4.57%           4.58%
SABAH                      3,110.39    7.62%           7.64%
SARAWAK                    3,103.08   10.76%          10.76%
JOHOR                      3,098.18   11.86%          11.83%
NEGERI SEMBILAN            3,072.16    5.98%           5.92%
SELANGOR                   3,070.86   10.69%          10.57%
KEDAH                      3,062.03    7.80%           7.69%
KUALA LUMPUR               3,058.17    0.23%           0.23%
LABUAN                     3,039.74    0.23%           0.23%
PERLIS                     3,004.91    1.33%           1.28%
PUTRAJAYA                  2,992.71    0.22%           0.21%

Enter State to see city-level detail (or press Enter to skip): Pahang
Enter the number of Top Cities to display (e.g., 5): 5


             ====================================================
                  Top 5 City-Level Detail for Pahang
                        For Years 2016 to 2025
             ====================================================

                                                                 Cumulative
 Rnk City                  Members   Avg Sales % Members  % Revenue  % Rev
---- -------------------- -------- ----------- ---------- ---------- ----------
   1 KUALA LIPIS                22    3,596.55    2.7%       3.1%       3.1%
   2 SUNGAI LEMBING             23    3,481.80    2.9%       3.2%       6.3%
   3 BUKIT FRASER               19    3,452.78    2.4%       2.6%       8.9%
   4 MENTAKAB                   22    3,409.16    2.7%       3.0%      11.8%
   5 BUKIT KUIN                 19    3,405.60    2.4%       2.6%      14.4%

Dropping temporary view...

Report complete.
```

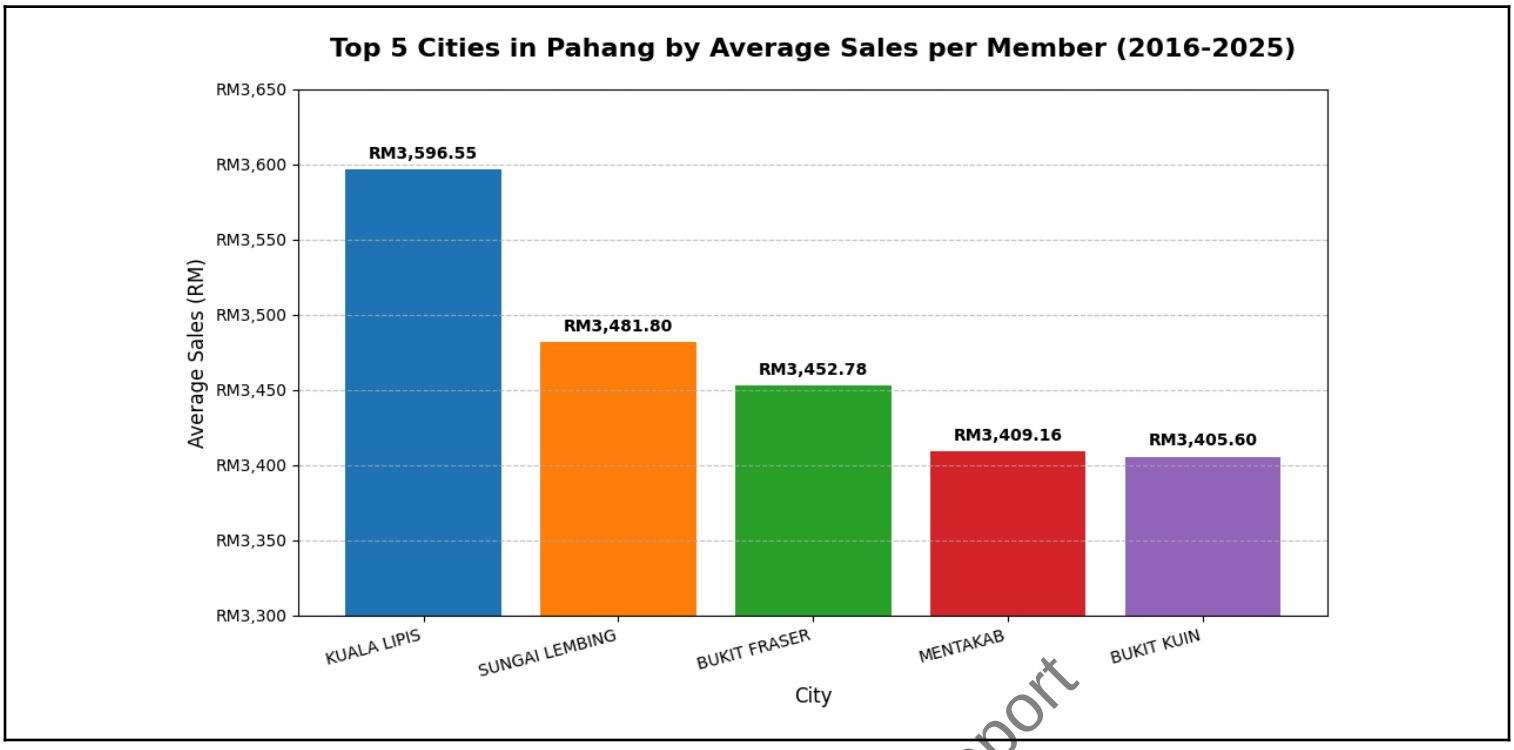
Sample Assignment Report

Visualisation:





Sample Assignment Report



Sample Assignment Report

Justification:

The report provides a performance breakdown by geography, showing where the library's most valuable members are from, over the 10 years. The '% of Total Revenue' shows a state's (or city's) contribution to the grand total sales across all states for the selected period. It is calculated using this formula: (State's Total Sales / Grand Total Sales for All States) * 100.

The library's most valuable members, on an individual basis, are from Pahang (RM 3,146.29 average sales per member) and Melaka (RM 3,143.16). This indicates that members from these states spend the most money per person. While not the highest in individual value, members from Perak form the largest single segment of the business, accounting for 16.12% of all active members and 16.27% of all revenue. Members who live in Kuala Lumpur contribute minimally to sales, accounting for only 0.23% of revenue and members. This indicates that the local academic community is not a significant source of revenue. Within the top-performing state of Pahang, revenue is driven by a small number of members from specific towns. For example, members from KUALA LIPIS and SUNGAI LEMBING are among the highest average spenders.

The high average sales figures (over RM 3,000) strongly imply that members from other states make large, consolidated purchases, such as buying all required textbooks for a semester at once. KL-based members (students, faculty, etc.) are likely commuters who have access to numerous large commercial bookstores in the city. These competitors may offer better pricing, discounts, or loyalty programs, drawing sales away from the campus library.

The library director should target the specific towns in Pahang that contain our highest-spending members, such as Kuala Lipis and Sungai Lembing. While the member count is low, their individual value is the highest. Organising smaller, personalized "pop-up" events or roadshows in these towns — perhaps hosted at local community halls, schools, or partner institutions — would allow the library to directly engage with these members. Activities could include exclusive previews of new academic titles, bundled textbook offers, and on-the-spot ordering services. By showing up in their community, the library signals that it values their loyalty and understands their academic needs, which can strengthen retention and encourage repeat purchases. Not only that, the library director should also focus on the state of our largest markets by volume: Perak and Johor. These states provide the majority of our revenue. In the future, a large-scale book fair just before the start of a new semester would be a major event, offering immense convenience. It would solidify our position as their primary supplier and capture sales that might otherwise go to local competitors.

Sample Assignment Report

3.2.3 Book Condition and Damage Analysis

SQL Code:

```
CLEAR BREAKS
CLEAR COMPUTES
CLEAR COLUMNS
SET VERIFY OFF
SET FEEDBACK OFF
SET PAGESIZE 50
SET LINESIZE 80

PROMPT
PROMPT Please provide the analysis period for the Book Damage Report.
ACCEPT start_year_prompt CHAR PROMPT 'Enter Start Year (e.g., 2016): '
ACCEPT end_year_prompt CHAR PROMPT 'Enter End Year (e.g., 2025): '
PROMPT

-- Report Section 1: Year/Season Breakdown with Damage Analysis
TTITLE CENTER '=========================================================' SKIP 1
-
       CENTER 'Loan Damage Analysis by Year and Season' SKIP 1 -
       CENTER 'For Years &start_year_prompt to &end_year_prompt' SKIP 1 -
       CENTER '=========================================================' SKIP 1
-
       LEFT 'Report Generated on: ' DATE  -
       RIGHT 'Page: ' SQL.PNO SKIP 2

COLUMN loan_year FORMAT 9999 HEADING 'Year'
COLUMN season FORMAT A12 HEADING 'Season'
COLUMN avg_daily_loans FORMAT 999,990.9 HEADING 'Avg |Daily Loans'
COLUMN pct_good_loans FORMAT A15 HEADING '% of Good Cond.|Books Loaned'
COLUMN pct_fair_loans FORMAT A15 HEADING '% of Fair Cond.|Books Loaned'
COLUMN pct_poor_loans FORMAT A15 HEADING '% of Poor Cond.|Books Loaned'

BREAK ON loan_year SKIP 1

WITH
-- Count days in each season for averaging
SEASON_DAYS AS (
    SELECT
        cal_year,
        CASE
            WHEN EXAM_WEEK_IND = 'Y' THEN 'Exam Week'
            WHEN STUDY_WEEK_IND = 'Y' THEN 'Study Week'
            WHEN HOLIDAY_IND = 'Y' THEN 'Holiday'
            ELSE 'Regular'
        END AS season,
        COUNT(DISTINCT date_key) AS num_days
```

Sample Assignment Report

```
    FROM DATE_DIM
    WHERE cal_year BETWEEN '&start_year_prompt' AND '&end_year_prompt'
    GROUP BY cal_year,
             CASE
                 WHEN EXAM_WEEK_IND = 'Y' THEN 'Exam Week'
                 WHEN STUDY_WEEK_IND = 'Y' THEN 'Study Week'
                 WHEN HOLIDAY_IND = 'Y' THEN 'Holiday'
                 ELSE 'Regular'
             END
),
-- Analyze all loans within the period, categorizing by current book
condition
LOAN_DAMAGE_ANALYSIS AS (
    SELECT
        dd.cal_year AS loan_year,
        CASE
            WHEN dd.EXAM_WEEK_IND = 'Y' THEN 'Exam Week'
            WHEN dd.STUDY_WEEK_IND = 'Y' THEN 'Study Week'
            WHEN dd.HOLIDAY_IND = 'Y' THEN 'Holiday'
            ELSE 'Regular'
        END AS season,
        COUNT(*) AS total_loans,
        SUM(CASE WHEN bd.bookCondition = 'GOOD' THEN 1 ELSE 0 END) AS
good_loans,
        SUM(CASE WHEN bd.bookCondition = 'FAIR' THEN 1 ELSE 0 END) AS
fair_loans,
        SUM(CASE WHEN bd.bookCondition = 'POOR' THEN 1 ELSE 0 END) AS
poor_loans
    FROM LOAN_FACT lf
    JOIN DATE_DIM dd ON lf.date_key = dd.date_key
    JOIN BOOK_DIM bd ON lf.book_key = bd.book_key
    WHERE dd.cal_year BETWEEN '&start_year_prompt' AND '&end_year_prompt'
    GROUP BY dd.cal_year,
             CASE
                 WHEN dd.EXAM_WEEK_IND = 'Y' THEN 'Exam Week'
                 WHEN dd.STUDY_WEEK_IND = 'Y' THEN 'Study Week'
                 WHEN dd.HOLIDAY_IND = 'Y' THEN 'Holiday'
                 ELSE 'Regular'
             END
)
-- Generate the final report with averages and percentages
SELECT
    lda.loan_year,
    lda.season,
    ROUND(lda.total_loans / sd.num_days, 1) AS avg_daily_loans,
    TO_CHAR(ROUND((lda.poor_loans * 100.0 / lda.total_loans), 1), '990.9')
|| '%' AS pct_poor_loans,
    TO_CHAR(ROUND((lda.fair_loans * 100.0 / lda.total_loans), 1), '990.9')
|| '%' AS pct_fair_loans,
```

Sample Assignment Report

```
    TO_CHAR(ROUND((lda.good_loans * 100.0 / lda.total_loans), 1), '990.9')
|| '%' AS pct_good_loans
FROM LOAN_DAMAGE_ANALYSIS lda
JOIN SEASON_DAYS sd ON lda.loan_year = sd.cal_year AND lda.season =
sd.season
ORDER BY lda.loan_year, pct_poor_loans DESC;

CLEAR BREAKS

-- Report Section 2: Interactive Drill-Down by Category
ACCEPT drilldown_year_prompt CHAR PROMPT 'Enter Year to Drill Down (e.g.,
2020): '
ACCEPT drilldown_season_prompt CHAR PROMPT 'Enter Season to Drill Down (Exam
Week/Study Week/Holiday/Regular): '

TTITLE CENTER '=====================================================' SKIP 1
-
       CENTER 'Category Damage Analysis for &drilldown_season_prompt
&drilldown_year_prompt' SKIP 1 -
       CENTER '=====================================================' SKIP 2

COLUMN book_category FORMAT A20 HEADING 'Book Category'
COLUMN pct_good_loaned FORMAT A15 HEADING '% of Good Cond.|Books Loaned'
COLUMN pct_fair_loaned FORMAT A15 HEADING '% of Fair Cond.|Books Loaned'
COLUMN pct_poor_loaned FORMAT A15 HEADING '% of Poor Cond.|Books Loaned'
COLUMN total_category_loans FORMAT '999,999' HEADING 'Total Loans'

WITH
CATEGORY_DAMAGE_ANALYSIS AS (
    SELECT
        bd.bookCategory,
        SUM(CASE WHEN bd.bookCondition = 'GOOD' THEN 1 ELSE 0 END) AS
good_books_loaned,
        SUM(CASE WHEN bd.bookCondition = 'FAIR' THEN 1 ELSE 0 END) AS
fair_books_loaned,
        SUM(CASE WHEN bd.bookCondition = 'POOR' THEN 1 ELSE 0 END) AS
poor_books_loaned,
        COUNT(*) AS total_category_loans
    FROM LOAN_FACT lf
    JOIN DATE_DIM dd ON lf.date_key = dd.date_key
    JOIN BOOK_DIM bd ON lf.book_key = bd.book_key
    WHERE dd.cal_year = '&drilldown_year_prompt'
      AND CASE
              WHEN dd.EXAM_WEEK_IND = 'Y' THEN 'EXAM WEEK'
              WHEN dd.STUDY_WEEK_IND = 'Y' THEN 'STUDY WEEK'
              WHEN dd.HOLIDAY_IND = 'Y' THEN 'HOLIDAY'
              ELSE 'REGULAR'
          END = UPPER('&drilldown_season_prompt')
    GROUP BY bd.bookCategory
```

Sample Assignment Report

```
)
SELECT
    bookCategory AS book_category,
    TO_CHAR(ROUND((poor_books_loaned * 100.0 / total_category_loans), 1),
'990.9') || '%' AS pct_poor_loaned,
    TO_CHAR(ROUND((fair_books_loaned * 100.0 / total_category_loans), 1),
'990.9') || '%' AS pct_fair_loaned,
    TO_CHAR(ROUND((good_books_loaned * 100.0 / total_category_loans), 1),
'990.9') || '%' AS pct_good_loaned,
    total_category_loans
FROM CATEGORY_DAMAGE_ANALYSIS
WHERE total_category_loans > 0
ORDER BY pct_poor_loaned DESC;

CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
UNDEFINE start_year_prompt
UNDEFINE end_year_prompt
UNDEFINE drilldown_year_prompt
UNDEFINE drilldown_season_prompt
SET FEEDBACK ON
SET VERIFY ON
TTITLE OFF
PROMPT
PROMPT Report complete.
PROMPT
```

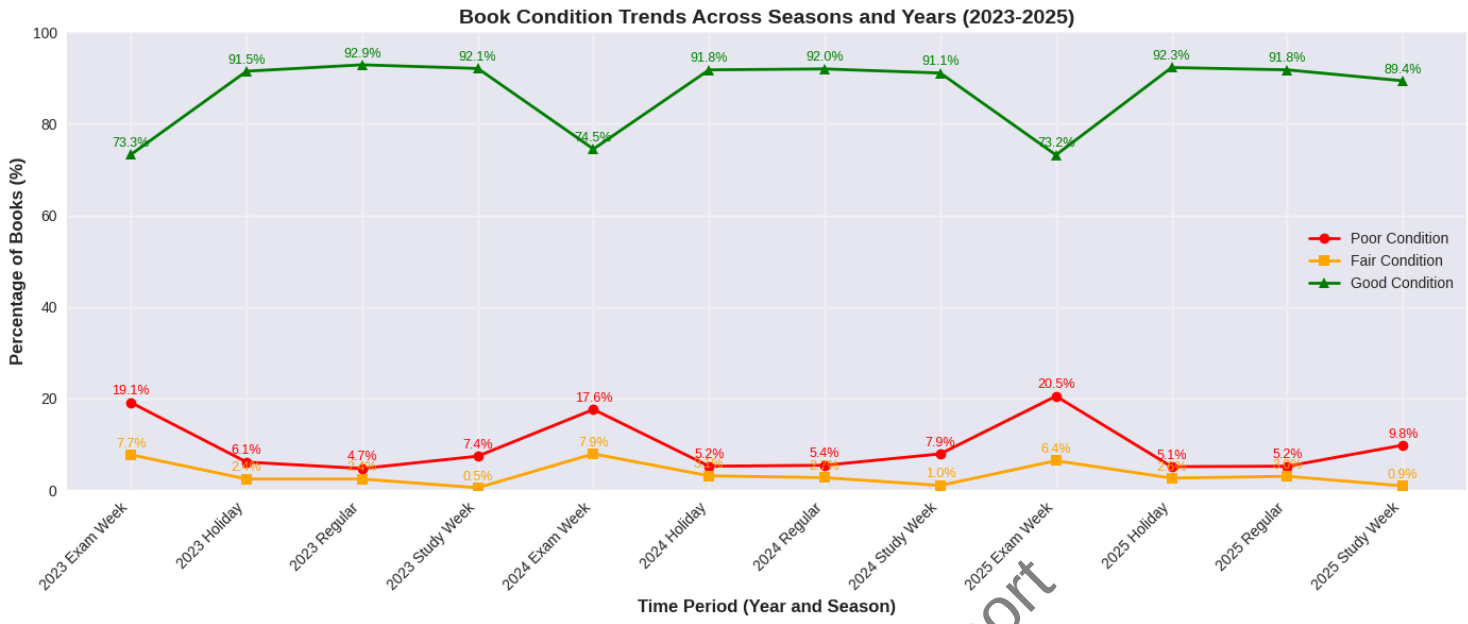
Sample Assignment Report

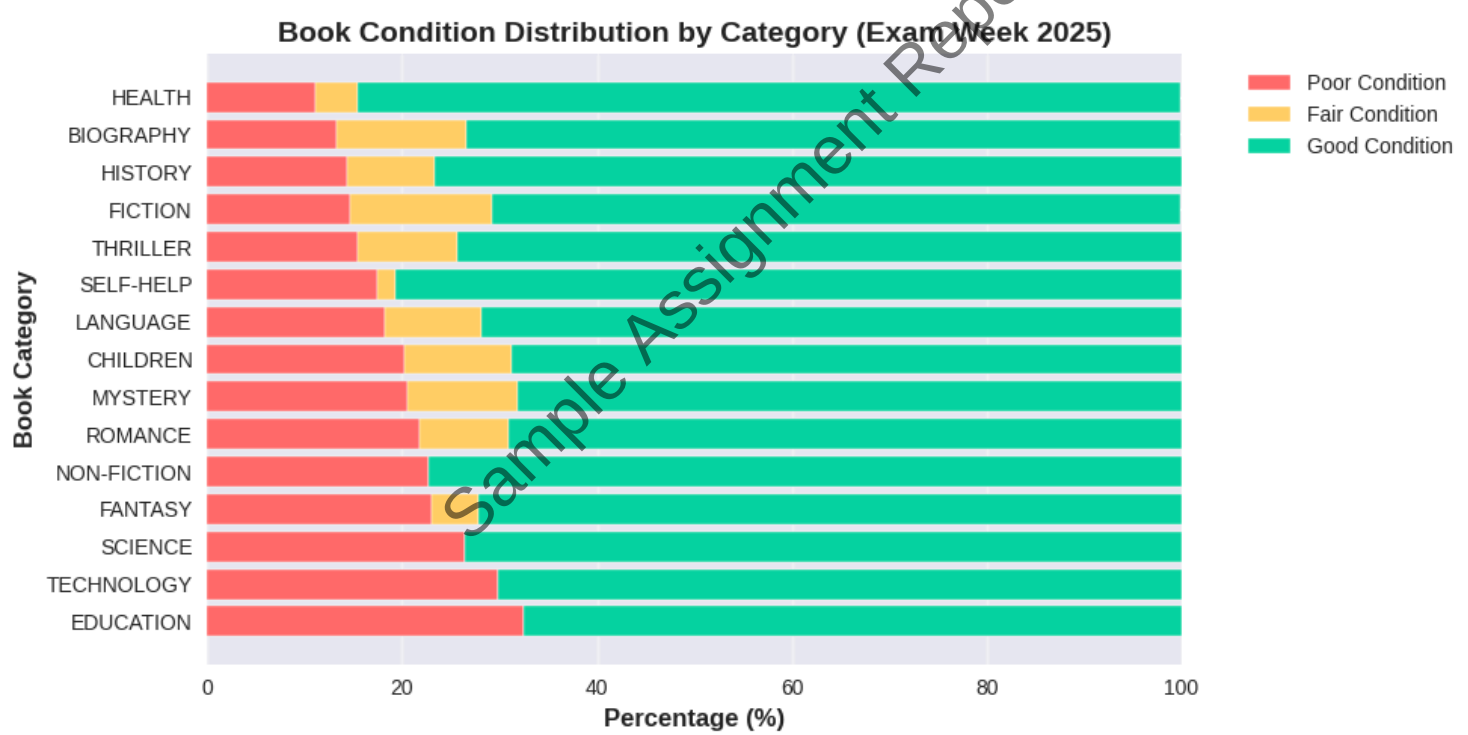
Output:

```
Please provide the analysis period for the Book Damage Report.
Enter Start Year (e.g., 2016): 2023
Enter End Year (e.g., 2025): 2025


              ====================================================
                     Loan Damage Analysis by Year and Season
                              For Years 2023 to 2025
              ====================================================
Report Generated on: 08-SEP-25                                  Page:          1

                          Avg  % of Poor Cond. % of Fair Cond. % of Good Cond.
 Year Season       Daily Loans Books Loaned    Books Loaned    Books Loaned
----- ------------ ----------- --------------- --------------- ---------------
 2023 Exam Week           26.8   19.1%            7.7%            73.3%
      Study Week          27.1    7.4%            0.5%            92.1%
      Holiday             27.2    6.1%            2.4%            91.5%
      Regular             27.4    4.7%            2.4%            92.9%

 2024 Exam Week           27.9   17.6%            7.9%            74.5%
      Study Week          28.4    7.9%            1.0%            91.1%
      Regular             27.2    5.4%            2.7%            92.0%
      Holiday             26.6    5.2%            3.1%            91.8%

 2025 Exam Week           26.9   20.5%            6.4%            73.2%
      Study Week          27.8    9.8%            0.9%            89.4%
      Regular             27.4    5.2%            3.0%            91.8%
      Holiday             29.1    5.1%            2.6%            92.3%

Enter Year to Drill Down (e.g., 2020): 2025
Enter Season to Drill Down (Exam Week/Study Week/Holiday/Regular): exam week

             ====================================================
                  Category Damage Analysis for exam week 2025
             ====================================================

                     % of Poor Cond. % of Fair Cond. % of Good Cond.
Book Category        Books Loaned    Books Loaned    Books Loaned    Total Loans
-------------------- --------------- --------------- --------------- -----------
EDUCATION              32.5%           0.0%            67.5%                 77
TECHNOLOGY             29.8%           0.0%            70.2%                 57
SCIENCE                26.5%           0.0%            73.5%                 49
FANTASY                23.1%           4.8%            72.1%                104
NON-FICTION            22.7%           0.0%            77.3%                 66
ROMANCE                21.8%           9.1%            69.1%                 55
MYSTERY                20.5%          11.4%            68.2%                 44
CHILDREN               20.3%          10.9%            68.8%                 64
LANGUAGE               18.3%           9.9%            71.8%                 71
SELF-HELP              17.5%           1.8%            80.7%                 57
THRILLER               15.4%          10.3%            74.4%                 78
FICTION                14.6%          14.6%            70.7%                 41
HISTORY                14.4%           8.9%            76.7%                 90
BIOGRAPHY              13.3%          13.3%            73.3%                 45
HEALTH                 11.1%           4.4%            84.4%                 45

Report complete.
```

Sample Assignment Report

Visualisation:





Sample Assignment Report

Justification:

This report provides a comprehensive book condition analysis by season (Regular, Study Week, Exam Week, and Holiday) of the past three years. The data reveals a clear and concerning pattern: book damage spikes dramatically during Exam Weeks. Across all three years, the percentage of books loaned out in "Poor Condition" is highest during this period, peaking at 20.5% in 2025. This is significantly higher than during Study Week (around 7-10%) and the Regular and Holiday seasons (around 4-6%). A detailed look into Exam Week of 2025 shows that the damage is most concentrated in specific academic categories. The top three most affected categories are: Education (32.5% in poor condition), Technology (29.8% in poor condition) and Science (26.5% in poor condition). This indicates that the books most essential for exam preparation are also the ones suffering the most wear and tear.

A consistent 17-20% damage rate during exam periods translates directly to higher operational costs. The library must allocate more of its budget to repairing and replacing books, particularly expensive academic textbooks. When a book is damaged, it's unavailable for other students. During the high-demand exam season, having one in five books in poor condition can severely limit access to crucial study materials, potentially impacting students' academic performance. The problem is not static; it's escalating. The rate of poor-condition loans during Exam Week has increased from 19.1% in 2023 to 20.5% in 2025. If this trend continues, the associated costs and availability issues will only worsen.

The data strongly suggests the root cause is the high-pressure academic environment of exam periods. Students are using academic books (like those in the Education, Technology, and Science categories) heavily for last-minute study. This intensive handling—flipping pages quickly, stuffing books into full backpacks, and long study sessions—accelerates wear and tear. Under stress, students may be less careful, leading to accidental damage like spills, torn pages, or broken spines. The issue likely compounds over time. Minor damage from one loan period is not repaired, and the book is loaned out again, accumulating more damage until it is finally classified as "Poor."

To address this issue, a multi-faceted approach is recommended. In the future, before and during exam seasons, launch a "Handle with Care" campaign. Use posters, social media, and bookmarks to remind students of the importance of preserving shared resources. Special emphasis should be placed on the most affected categories — Education, Technology, and Science — with targeted slogans like “Your Education books educate everyone — keep them in top shape”. Moreover, using the historical data, librarians can identify the specific titles and categories most at risk. A structured “pre-exam maintenance week” could be scheduled, during which staff inspect these books for early signs of wear — loose bindings, torn pages, or water damage. Minor repairs (e.g., spine reinforcement, page mending) can be done immediately to extend the book’s usable life. This proactive approach ensures that the most in-demand books are in good condition when students need them most, reducing the likelihood of further damage during peak usage. For resource management, library directors should consider allocating a portion of the acquisitions budget specifically for replacing high-use books in the top-damaged categories, treating it as a predictable operational cost. For instance, if the annual acquisitions budget is RM 200,000, allocating 10% (RM 20,000) to replacing high-use books in Education, Technology, and Science ensures that funds are always available to maintain the quality of the most critical resources. This allocation could be reviewed annually and adjusted based on updated damage statistics, ensuring the budget remains aligned with actual needs.

Sample Assignment Report

3.3 Lai Zheng Xuan

3.3.1 Loan Behavior and Overdue Fine Analysis

SQL Code:

```
--
-------------------------------------------------------------------------------
-------
-- SQL*Plus Environment Setup
--
-------------------------------------------------------------------------------
-------
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
SET PAGESIZE 35
SET LINESIZE 130
SET FEEDBACK OFF
SET VERIFY OFF
SET TERMOUT ON

--
-------------------------------------------------------------------------------
-------
-- User Input Variables
--
-------------------------------------------------------------------------------
-------
COLUMN year_col NEW_VALUE P_END_YEAR_DEFAULT
COLUMN start_year_col NEW_VALUE P_START_YEAR_DEFAULT

SELECT
    TO_CHAR(SYSDATE, 'YYYY') - 1 AS year_col,
    TO_CHAR(SYSDATE, 'YYYY') - 5 AS start_year_col
FROM DUAL;

DEFINE P_QUARTER_DEFAULT = 'ALL'

SPOOL "C:\DW assignment\R1_Enhanced_result.txt"

ACCEPT P_START_YEAR NUMBER DEFAULT &P_START_YEAR_DEFAULT PROMPT 'Enter Start
Analysis Year (e.g., &P_START_YEAR_DEFAULT): '
ACCEPT P_END_YEAR NUMBER DEFAULT &P_END_YEAR_DEFAULT PROMPT 'Enter End
Analysis Year (e.g., &P_END_YEAR_DEFAULT): '
ACCEPT P_QUARTER CHAR DEFAULT '&P_QUARTER_DEFAULT' PROMPT 'Enter Quarter
(1-4, or &P_QUARTER_DEFAULT for all quarters): '

--
```

Sample Assignment Report

```
==============================================================================
=======
-- VIEW CREATION
--
==============================================================================
=======
CREATE OR REPLACE VIEW V_LOAN_ANALYSIS AS
SELECT
    lf.loanID,
    dd.CAL_YEAR,
    dd.CAL_QUARTER,
    bd.bookCategory,
    md.memberID,
    CASE
        WHEN FLOOR(MONTHS_BETWEEN(dd.CAL_DATE, md.dob)/12) < 25 THEN 'Young
(U25)'
        WHEN FLOOR(MONTHS_BETWEEN(dd.CAL_DATE, md.dob)/12) BETWEEN 25 AND 40
THEN 'Middle (25-40)'
        WHEN FLOOR(MONTHS_BETWEEN(dd.CAL_DATE, md.dob)/12) BETWEEN 41 AND 60
THEN 'Senior (41-60)'
        ELSE 'Elder (>60)'
    END AS MemberAgeGroup,
    lf.overdueDays,
    lf.totalFine,
    CASE WHEN lf.overdueDays > 0 THEN 1 ELSE 0 END AS IsOverdue,
    lf.fine_paid_flag AS IsFinePaid
FROM
    LOAN_FACT lf
JOIN
    DATE_DIM dd ON lf.date_key = dd.DATE_KEY
JOIN
    BOOK_DIM bd ON lf.book_key = bd.book_key
JOIN
    MEMBER_DIM md ON lf.member_key = md.member_key
/

--
==============================================================================
=======
-- REPORT SECTION 1: OVERALL PERFORMANCE SUMMARY (for the entire date range)
--
==============================================================================
=======
TTITLE CENTER '-------------------------------------------------------' SKIP
1 -
CENTER ' LOAN BEHAVIOR AND OVERDUE FINE ANALYSIS: ' SKIP 1 -
CENTER ' OVERALL PERFORMANCE SUMMARY ' SKIP 1 -
CENTER ' YEARS: &P_START_YEAR - &P_END_YEAR | QUARTER: &P_QUARTER ' SKIP 1 -
CENTER '-------------------------------------------------------' SKIP 1 -
```

Sample Assignment Report

```
LEFT 'DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN TotalLoans        HEADING 'Total Loan    '                FORMAT
999,990
COLUMN OverdueLoans      HEADING 'Overdue Loan    '              FORMAT
999,990
COLUMN FinesGenerated    HEADING 'Total Fine Generated(RM)'      FORMAT
999,999.00
COLUMN FinesCollected    HEADING 'Total Fine Collected(RM)'      FORMAT
999,999.00
COLUMN OverdueRate       HEADING '    Overdue Rate(%)'           FORMAT
990.00
COLUMN CollectionEff     HEADING '    Collection Efficiency(%)' FORMAT
990.00

SELECT
    COUNT(loanID) AS TotalLoans,
    SUM(IsOverdue) AS OverdueLoans,
    SUM(totalFine) AS FinesGenerated,
    SUM(CASE WHEN IsFinePaid = 1 THEN totalFine ELSE 0 END) AS
FinesCollected,
    (SUM(IsOverdue) / NULLIF(COUNT(loanID), 0)) * 100 AS OverdueRate,
    (SUM(CASE WHEN IsFinePaid = 1 THEN totalFine ELSE 0 END) /
NULLIF(SUM(totalFine), 0)) * 100 AS CollectionEff
FROM
    V_LOAN_ANALYSIS
WHERE
    CAL_YEAR BETWEEN &P_START_YEAR AND &P_END_YEAR
    AND (CAL_QUARTER = '&P_QUARTER' OR UPPER('&P_QUARTER') = 'ALL');

CLEAR COMPUTES;

--
==========================================================================
=======
-- REPORT SECTION 2: PERFORMANCE BY TOP 5 CATEGORIES AND SEGMENTS (for the
entire date range)
--
==========================================================================
=======
TTITLE CENTER
'-----------------------------------------------------------------------' SKIP
1 -
CENTER ' LOAN BEHAVIOR AND OVERDUE FINE ANALYSIS: ' SKIP 1 -
CENTER ' PERFORMANCE BY TOP 5 BOOK CATEGORIES AND MEMBER SEGMENTS ' SKIP 1 -
CENTER ' YEARS: &P_START_YEAR - &P_END_YEAR | QUARTER: &P_QUARTER ' SKIP 1 -
CENTER
'-----------------------------------------------------------------------' SKIP
1 -
```

Sample Assignment Report

```
LEFT 'DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON bookCategory SKIP 1 ON REPORT
COMPUTE SUM LABEL 'Total' OF TotalLoans OverdueLoans FinesGenerated
FinesCollected ON bookCategory
COMPUTE SUM LABEL 'Grand Total' OF TotalLoans OverdueLoans FinesGenerated
FinesCollected ON REPORT

COLUMN bookCategory      HEADING 'Book Category'    FORMAT A15
COLUMN MemberAgeGroup    HEADING 'Age Group'        FORMAT A20
COLUMN TotalLoans        HEADING '    Loan'         FORMAT 999,990
COLUMN OverdueLoans      HEADING '    Overdue'      FORMAT 999,990
COLUMN FinesGenerated    HEADING 'Fine Gen.(RM)'    FORMAT 999,999.00
COLUMN FinesCollected    HEADING 'Fine Coll.(RM)'   FORMAT 999,999.00
COLUMN OverdueRate       HEADING '  Overdue(%)'     FORMAT 990.00
COLUMN CollectionEff     HEADING '  Collection(%)'  FORMAT 990.00
COLUMN FineContribution  HEADING '  Fine Con.(%)'   FORMAT 990.00

WITH AggregatedData AS (
    SELECT
        v.bookCategory,
        v.MemberAgeGroup,
        COUNT(v.loanID) AS TotalLoans,
        SUM(v.IsOverdue) AS OverdueLoans,
        SUM(v.totalFine) AS FinesGenerated,
        SUM(CASE WHEN v.IsFinePaid = 1 THEN v.totalFine ELSE 0 END) AS
FinesCollected,
        (SUM(v.IsOverdue) / NULLIF(COUNT(v.loanID), 0)) * 100 AS
OverdueRate,
        (SUM(CASE WHEN v.IsFinePaid = 1 THEN v.totalFine ELSE 0 END) /
NULLIF(SUM(v.totalFine), 0)) * 100 AS CollectionEff,
        (SUM(v.totalFine) * 100) / SUM(SUM(v.totalFine)) OVER () AS
FineContribution,
        SUM(SUM(v.totalFine)) OVER (PARTITION BY v.bookCategory) as
TotalFinesPerCategory
    FROM
        V_LOAN_ANALYSIS v
    WHERE
        v.CAL_YEAR BETWEEN &P_START_YEAR AND &P_END_YEAR
        AND (v.CAL_QUARTER = '&P_QUARTER' OR UPPER('&P_QUARTER') = 'ALL')
    GROUP BY
        v.bookCategory, v.MemberAgeGroup
),
RankedData AS (
    SELECT
        d.*,
        DENSE_RANK() OVER (ORDER BY d.TotalFinesPerCategory DESC) as
CategoryRank
    FROM
```

Sample Assignment Report

```
        AggregatedData d
)
SELECT
    bookCategory,
    MemberAgeGroup,
    TotalLoans,
    OverdueLoans,
    FinesGenerated,
    FinesCollected,
    OverdueRate,
    CollectionEff,
    FineContribution
FROM
    RankedData
WHERE
    CategoryRank <= 5
ORDER BY
    CategoryRank,
    bookCategory,
    CASE MemberAgeGroup
        WHEN 'Young (U25)'    THEN 1
        WHEN 'Middle (25-40)' THEN 2
        WHEN 'Senior (41-60)' THEN 3
        WHEN 'Elder (>60)'    THEN 4
        ELSE 5
    END;

CLEAR BREAKS;
CLEAR COMPUTES;

--
================================================================================
=======
-- REPORT SECTION 3: MULTI-YEAR TREND ANALYSIS (Pivoted View with Growth)
--
================================================================================
=======
TTITLE CENTER
'-------------------------------------------------------------------------------
---------------------' SKIP 1 -
CENTER ' LOAN BEHAVIOR AND OVERDUE FINE ANALYSIS: ' SKIP 1 -
CENTER ' MULTI-YEAR TREND ANALYSIS IN ' SKIP 1 -
CENTER ' YEARS: &P_START_YEAR - &P_END_YEAR ' SKIP 1 -
CENTER
'-------------------------------------------------------------------------------
---------------------' SKIP 1 -
LEFT 'DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

CLEAR BREAKS
```

Sample Assignment Report

```
COMPUTE SUM LABEL 'Grand Total' OF Q1_Loans Q1_Fines Q2_Loans Q2_Fines
Q3_Loans Q3_Fines Q4_Loans Q4_Fines Total_Loans Total_Fines ON REPORT

COLUMN CAL_YEAR          HEADING 'Year'                  FORMAT 9999
COLUMN Q1_Loans          HEADING '  Q1|Loans'            FORMAT 999,990
COLUMN Q1_Fines          HEADING 'Q1|Fines(RM)'          FORMAT 99,999.00
COLUMN Q2_Loans          HEADING 'Q2|Loans'              FORMAT 999,990
COLUMN Q2_Fines          HEADING 'Q2|Fines(RM)'          FORMAT 99,999.00
COLUMN Q3_Loans          HEADING 'Q3|Loans'              FORMAT 999,990
COLUMN Q3_Fines          HEADING 'Q3|Fines(RM)'          FORMAT 99,999.00
COLUMN Q4_Loans          HEADING 'Q4|Loans'              FORMAT 999,990
COLUMN Q4_Fines          HEADING 'Q4|Fines(RM)'          FORMAT 99,999.00
COLUMN Total_Loans       HEADING 'Total|    Loans'       FORMAT 999,990
COLUMN Total_Fines       HEADING 'Total|   Fines(RM)'    FORMAT 999,999.00
COLUMN LoanGrowth_Pct    HEADING 'Loan  |  Growth %'     FORMAT 990.00
COLUMN FineGrowth_Pct    HEADING 'Fine  |  Growth %'     FORMAT 990.00

WITH PivotedData AS (
    SELECT
        CAL_YEAR,
        COUNT(CASE WHEN CAL_QUARTER = 1 THEN loanID END) AS Q1_Loans,
        SUM(CASE WHEN CAL_QUARTER = 1 THEN totalFine ELSE 0 END) AS
Q1_Fines,
        COUNT(CASE WHEN CAL_QUARTER = 2 THEN loanID END) AS Q2_Loans,
        SUM(CASE WHEN CAL_QUARTER = 2 THEN totalFine ELSE 0 END) AS
Q2_Fines,
        COUNT(CASE WHEN CAL_QUARTER = 3 THEN loanID END) AS Q3_Loans,
        SUM(CASE WHEN CAL_QUARTER = 3 THEN totalFine ELSE 0 END) AS
Q3_Fines,
        COUNT(CASE WHEN CAL_QUARTER = 4 THEN loanID END) AS Q4_Loans,
        SUM(CASE WHEN CAL_QUARTER = 4 THEN totalFine ELSE 0 END) AS
Q4_Fines,
        COUNT(loanID) AS Total_Loans,
        SUM(totalFine) AS Total_Fines
    FROM
        V_LOAN_ANALYSIS
    WHERE
        CAL_YEAR BETWEEN &P_START_YEAR AND &P_END_YEAR
        AND ('&P_QUARTER' = 'ALL' OR CAL_QUARTER = TO_NUMBER('&P_QUARTER'))
    GROUP BY
        CAL_YEAR
)
SELECT
    p.CAL_YEAR,
    p.Q1_Loans, p.Q1_Fines,
    p.Q2_Loans, p.Q2_Fines,
    p.Q3_Loans, p.Q3_Fines,
    p.Q4_Loans, p.Q4_Fines,
```

Sample Assignment Report

```
    p.Total_Loans,
    p.Total_Fines,
    -- Calculate Loan Growth %
    ( (p.Total_Loans - LAG(p.Total_Loans, 1, 0) OVER (ORDER BY p.CAL_YEAR))
/ NULLIF(LAG(p.Total_Loans, 1, 0) OVER (ORDER BY p.CAL_YEAR), 0) ) * 100 AS
LoanGrowth_Pct,
    -- Calculate Fine Growth %
    ( (p.Total_Fines - LAG(p.Total_Fines, 1, 0) OVER (ORDER BY p.CAL_YEAR))
/ NULLIF(LAG(p.Total_Fines, 1, 0) OVER (ORDER BY p.CAL_YEAR), 0) ) * 100 AS
FineGrowth_Pct
FROM
    PivotedData p
ORDER BY
    p.CAL_YEAR;

--
================================================================================
=======
-- Final Cleanup
--
================================================================================
=======
SPOOL OFF;
TTITLE OFF;
CLEAR COLUMNS;
CLEAR BREAKS;
CLEAR COMPUTES;
SET TERMOUT ON;
SET FEEDBACK ON;
SET VERIFY ON;

UNDEFINE P_START_YEAR;
UNDEFINE P_END_YEAR;
UNDEFINE P_QUARTER;
```

Sample Assignment Report

Output:

```
Enter Start Analysis Year (e.g.,       2023): 2020
Enter End Analysis Year (e.g., 2025): 2024
Enter Quarter (1-4, or ALL for all quarters): ALL

                                    -------------------------------------------------------
                                           LOAN BEHAVIOR AND OVERDUE FINE ANALYSIS:
                                                   OVERALL PERFORMANCE SUMMARY
                                        YEARS:       2020 -       2024 | QUARTER: ALL
                                    -------------------------------------------------------
DATE: 18-SEP-25                                                                                                                                PAGE:    1

Total Loan     Overdue Loan     Total Fine Generated(RM) Total Fine Collected(RM)     Overdue Rate(%)     Collection Efficiency(%)
-------------- ---------------- ------------------------ ------------------------ ------------------- ----------------------------
        50,104           15,016               561,630.00               393,470.00               29.97                        70.06


                                 ---------------------------------------------------------------------
                                           LOAN BEHAVIOR AND OVERDUE FINE ANALYSIS:
                                     PERFORMANCE BY TOP 5 BOOK CATEGORIES AND MEMBER SEGMENTS
                                        YEARS:       2020 -       2024 | QUARTER: ALL
                                 ---------------------------------------------------------------------
DATE: 18-SEP-25                                                                                                                                PAGE:    1

Book Category   Age Group                Loan     Overdue Fine Gen.(RM) Fine Coll.(RM)   Overdue(%)   Collection(%)   Fine Con.(%)
--------------- -------------------- -------- ---------- ------------- ------------- ------------ --------------- -------------
FANTASY         Young (U25)               184         51      2,060.00      1,355.00        27.72           65.78          0.37
                Middle (25-40)          1,314        394     15,490.00     10,760.00        29.98           69.46          2.76
                Senior (41-60)          2,584        795     29,820.00     20,085.00        30.77           67.35          5.31
                Elder (>60)               884        275     10,355.00      7,450.00        31.11           71.95          1.84
***************                      -------- ---------- ------------- -------------
Total                                   4,966      1,515     57,725.00     39,650.00

HISTORY         Young (U25)               195         68      2,275.00      1,610.00        34.87           70.77          0.41
                Middle (25-40)          1,233        359     12,770.00      9,645.00        29.12           75.53          2.27
                Senior (41-60)          2,424        746     27,785.00     18,245.00        30.78           65.66          4.95
                Elder (>60)               886        267     10,205.00      7,225.00        30.14           70.80          1.82
***************                      -------- ---------- ------------- -------------
Total                                   4,738      1,440     53,035.00     36,725.00

EDUCATION       Young (U25)               151         58      2,520.00      1,720.00        38.41           68.25          0.45
                Middle (25-40)          1,259        353     13,310.00      9,395.00        28.04           70.59          2.37
                Senior (41-60)          2,211        631     24,405.00     17,550.00        28.54           71.91          4.35
                Elder (>60)               801        241      9,015.00      6,645.00        30.09           73.71          1.61
***************                      -------- ---------- ------------- -------------
Total                                   4,422      1,283     49,250.00     35,310.00

NON-FICTION     Young (U25)               158         50      1,955.00      1,480.00        31.65           75.70          0.35
                Middle (25-40)          1,017        316     11,635.00      8,480.00        31.07           72.88          2.07
                Senior (41-60)          1,930        604     23,115.00     16,135.00        31.30           69.80          4.12
                Elder (>60)               647        217      7,945.00      5,800.00        33.54           73.00          1.41
```

Sample Assignment Report

LOAN BEHAVIOR AND OVERDUE FINE ANALYSIS:
PERFORMANCE BY TOP 5 BOOK CATEGORIES AND MEMBER SEGMENTS
YEARS: 2020 - 2024 | QUARTER: ALL

DATE: 18-SEP-25 PAGE: 2

Book Category	Age Group	Loan	Overdue	Fine Gen.(RM)	Fine Coll.(RM)	Overdue(%)	Collection(%)	Fine Con.(%)
**********								
Total		3,752	1,187	44,650.00	31,895.00			
THRILLER	Young (U25)	135	35	1,210.00	830.00	25.93	68.60	0.22
	Middle (25-40)	979	295	11,175.00	7,645.00	30.13	68.41	1.99
	Senior (41-60)	1,949	586	21,625.00	16,210.00	30.07	74.96	3.85
	Elder (>60)	689	214	7,560.00	4,985.00	31.06	65.94	1.35
**********								
Total		3,752	1,130	41,570.00	29,670.00			
Grand Total		21,630	6,555	246,230.00	173,250.00			

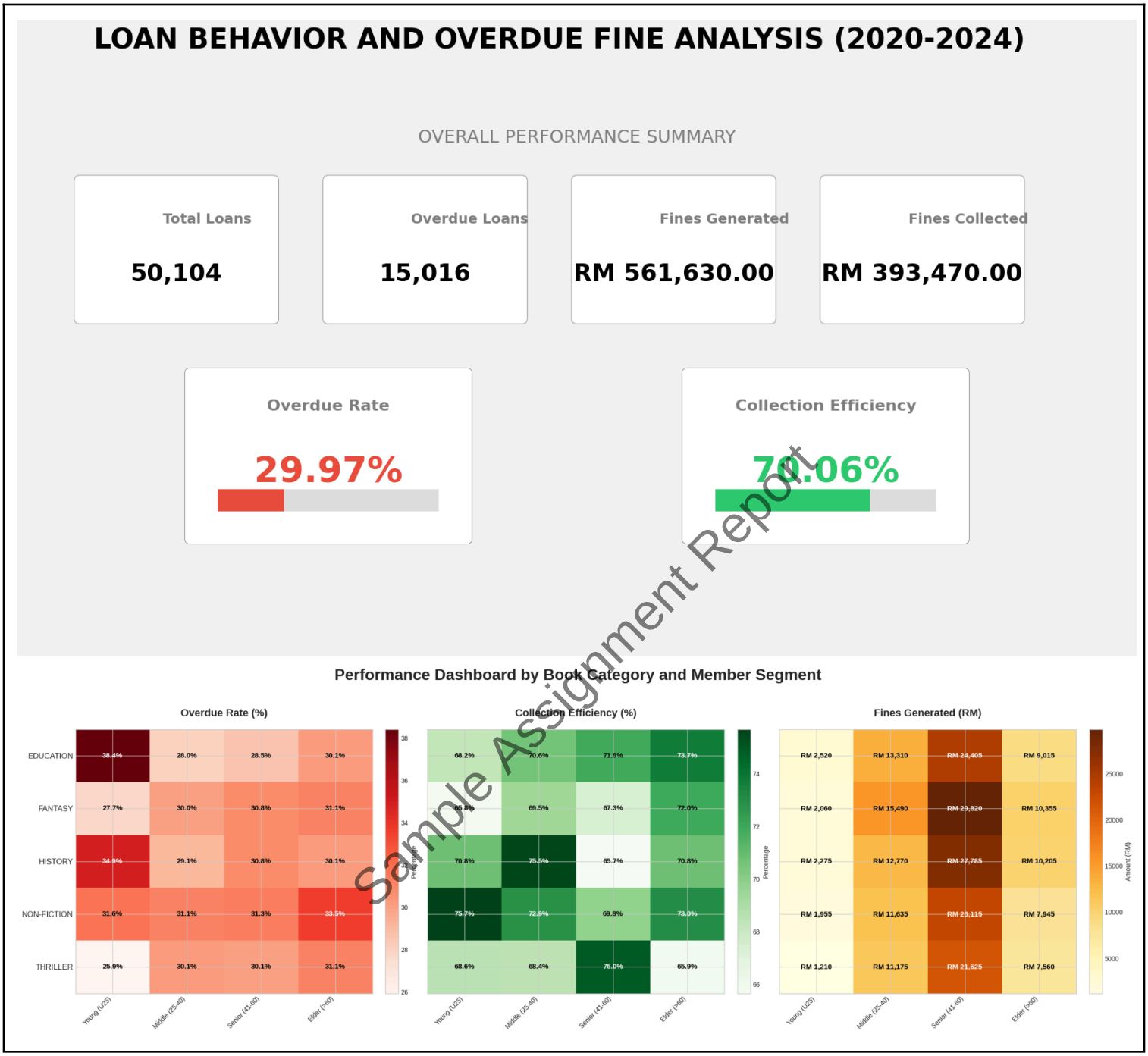
LOAN BEHAVIOR AND OVERDUE FINE ANALYSIS:
MULTI-YEAR TREND ANALYSIS IN
YEARS: 2020 - 2024

DATE: 18-SEP-25 PAGE: 1

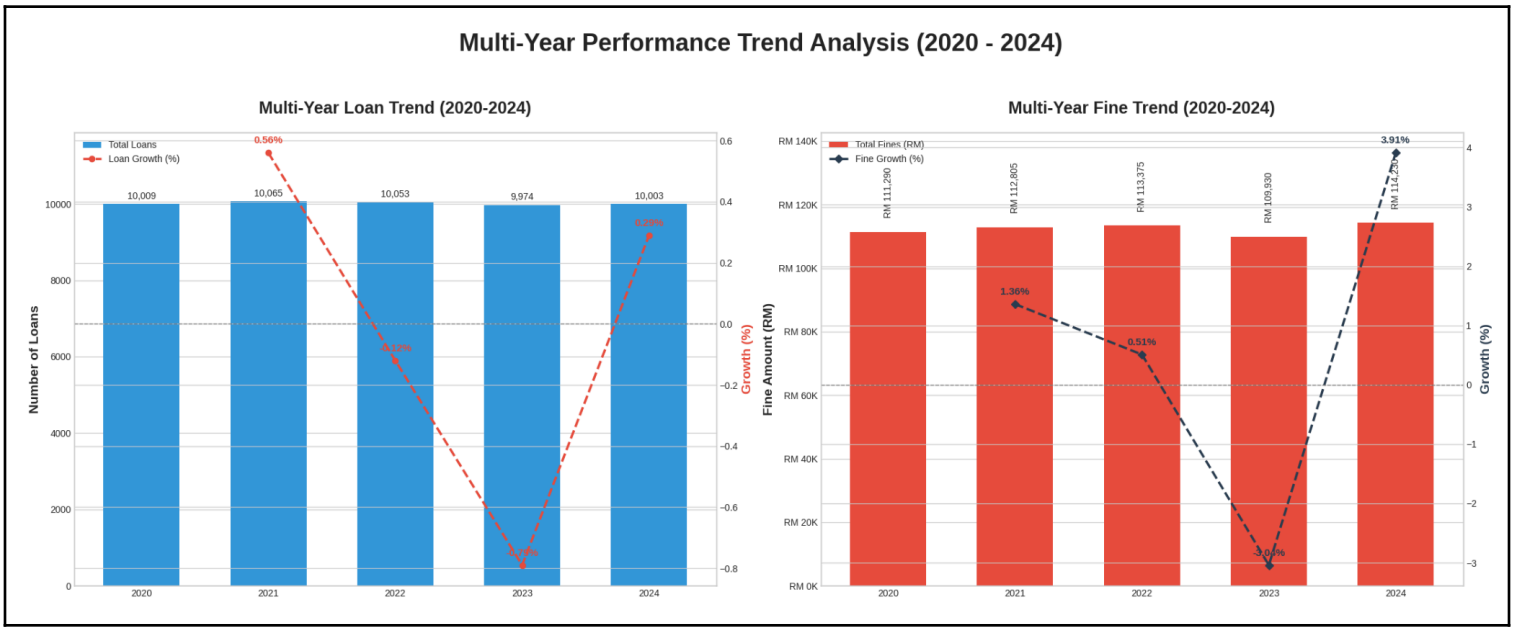
Year	Q1 Loans	Q1 Fines(RM)	Q2 Loans	Q2 Fines(RM)	Q3 Loans	Q3 Fines(RM)	Q4 Loans	Q4 Fines(RM)	Total Loans	Total Fines(RM)	Loan Growth %	Fine Growth %
2020	2,553	27,525.00	2,450	27,235.00	2,442	27,125.00	2,564	29,405.00	10,009	111,290.00		
2021	2,476	26,670.00	2,546	28,785.00	2,645	29,605.00	2,398	27,745.00	10,065	112,805.00	0.56	1.36
2022	2,457	26,995.00	2,508	30,070.00	2,545	27,885.00	2,543	28,425.00	10,053	113,375.00	-0.12	0.51
2023	2,493	27,510.00	2,483	27,880.00	2,502	27,800.00	2,496	26,740.00	9,974	109,930.00	-0.79	-3.04
2024	2,426	27,860.00	2,473	28,445.00	2,533	28,400.00	2,571	29,525.00	10,003	114,230.00	0.29	3.91

Sample Assignment Report

Visualisation:



Sample Assignment Report



Justification:

This report which focuses on loan behaviour and overdue fine analysis is made up of 3 sections. It shows an overall performance summary of loan behaviour and overdue fine from year 2020 to year 2024, performance by top 5 book categories and member segments from year 2020 to year 2024, and Year-Over-Year performance comparison between 2024 and 2025.

Sample Assignment Report

Overall Performance Summary

The first section of the report shows that from 2020 to 2024, the library loaned out 50,104 books, but 15,016(29.97%) of them were returned late. This is a problem because other members have to wait longer for books they want to read. The library charged over RM 561,630.00 in late fees, but only succeeded in collecting RM 393,470.00 (70.06%) of that money. This means the library is losing income. To fix this, I suggest two simple actions.

1. First, send friendly email or text reminders to members a few days before their books are due.
2. Second, if a member has an unpaid fine, they should not be allowed to borrow more books until they pay it.

Performance by Book Category and Member Segment

The second section tells us exactly where the risk of overdue loans is concentrated. There are 18 categories of book but this section only shows the top 5 categories with the most fine created. By looking at which books and which groups of people have the most late fees, we can see some clear patterns. Younger members (under 25) consistently demonstrate the highest overdue rates, particularly in categories like Education (38.4%) and History (34.87%). This suggests a need for targeted interventions for this demographic. On the other hand, Senior members (41-60), while having a moderate overdue rate, are the most active borrowers. Consequently, they generate the largest volume of fines and represent the most significant portion of the library's fine-related income. The library's success in collecting these fines can vary a lot, which tells us that having the same rules for everyone isn't working. We need different approaches for different people.

1. For younger members, consider shorter loan periods for popular items or limiting the number of books they

can borrow at one time from high-risk categories.

2. For our most active borrowers, the seniors, we could offer a paid membership with benefits like longer borrowing times, which would reward them and create a new source of income for the library.
3. Fine collection efforts should be prioritised on the middle-aged and senior groups, as they account for the largest amount of generated fines, offering the greatest potential for financial return.

Multi-Year Trend Analysis

The third section looks at the library's performance from 2020 to 2024. It shows that both the number of loans and the amount of money from fines have stayed mostly the same, with only small changes each year. This tells us that people's borrowing habits aren't changing, and the problem with late books is not going away on its own.Since the number of loans isn't growing, it means the library isn't attracting many new members. Also, the money from fines goes up and down, which makes it an unreliable way to get income. This shows that the library needs to take action. We cannot just wait for things to get better or depend on fine money to keep the library running. It is very important to use the solutions mentioned earlier and to look for new ways to make money, like holding paid workshops, events, or selling more books. This will help make sure the library has enough money to operate in the future.

Sample Assignment Report

3.3.2 Peak Loan and Reservation Date and Staffing Optimisation

SQL Code:

```
--
-------------------------------------------------------------------------------
-------
-- SQL*Plus Environment Setup
--
-------------------------------------------------------------------------------
-------
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
SET PAGESIZE 35
SET LINESIZE 130
SET FEEDBACK OFF
SET VERIFY OFF
SET TERMOUT ON

--
-------------------------------------------------------------------------------
-------
-- User Input Variables (Enhanced for Date Range)
--
-------------------------------------------------------------------------------
-------
COLUMN end_year_col NEW_VALUE P_END_YEAR_DEFAULT
COLUMN start_year_col NEW_VALUE P_START_YEAR_DEFAULT

SELECT
    TO_CHAR(SYSDATE, 'YYYY') - 1 AS end_year_col,
    TO_CHAR(SYSDATE, 'YYYY') - 5 AS start_year_col
FROM DUAL;

DEFINE P_MONTH_DEFAULT = 'ALL'

SPOOL "C:\DW assignment\R2_Enhanced_result_v2.txt"

ACCEPT P_START_YEAR NUMBER DEFAULT &P_START_YEAR_DEFAULT PROMPT 'Enter Start
Analysis Year (e.g., &P_START_YEAR_DEFAULT): '
ACCEPT P_END_YEAR NUMBER DEFAULT &P_END_YEAR_DEFAULT PROMPT 'Enter End
Analysis Year (e.g., &P_END_YEAR_DEFAULT): '
ACCEPT P_MONTH CHAR DEFAULT '&P_MONTH_DEFAULT' PROMPT 'Enter Month Name
(e.g., JANUARY, or ALL for all months): '

--
===============================================================================
=======
```

Sample Assignment Report

```
-- VIEW CREATION
--
==========================================================================
=======
CREATE OR REPLACE VIEW V_ACTIVITY_ANALYSIS AS
SELECT
    lf.loanID AS ActivityID,
    'Loan' AS ActivityType,
    lf.staff_key,
    sd.department,
    sd.position,
    dd.cal_year, dd.cal_month_name, dd.cal_date, dd.day_week,
dd.day_num_month,
    ((TRUNC(dd.cal_date, 'IW') - TRUNC(TRUNC(dd.cal_date, 'MM'), 'IW')) / 7)
+ 1 AS WeekOfMonth,
    CASE WHEN UPPER(SUBSTR(dd.day_week, 1, 3)) IN ('SAT', 'SUN') THEN
'Weekend' ELSE 'Weekday' END AS DayType,
    CASE UPPER(SUBSTR(dd.day_week, 1, 3))
        WHEN 'MON' THEN '1-MONDAY' WHEN 'TUE' THEN '2-TUESDAY' WHEN 'WED'
THEN '3-WEDNESDAY'
        WHEN 'THU' THEN '4-THURSDAY' WHEN 'FRI' THEN '5-FRIDAY' WHEN 'SAT'
THEN '6-SATURDAY'
        WHEN 'SUN' THEN '7-SUNDAY'
    END AS DaySortKey
FROM LOAN_FACT lf
JOIN DATE_DIM dd ON lf.date_key = dd.date_key
JOIN STAFF_DIM sd ON lf.staff_key = sd.staff_key
UNION ALL
SELECT
    rf.reserveID AS ActivityID,
    'Reservation' AS ActivityType,
    rf.staff_key,
    sd.department,
    sd.position,
    dd.cal_year, dd.cal_month_name, dd.cal_date, dd.day_week,
dd.day_num_month,
    ((TRUNC(dd.cal_date, 'IW') - TRUNC(TRUNC(dd.cal_date, 'MM'), 'IW')) / 7)
+ 1 AS WeekOfMonth,
    CASE WHEN UPPER(SUBSTR(dd.day_week, 1, 3)) IN ('SAT', 'SUN') THEN
'Weekend' ELSE 'Weekday' END AS DayType,
    CASE UPPER(SUBSTR(dd.day_week, 1, 3))
        WHEN 'MON' THEN '1-MONDAY' WHEN 'TUE' THEN '2-TUESDAY' WHEN 'WED'
THEN '3-WEDNESDAY'
        WHEN 'THU' THEN '4-THURSDAY' WHEN 'FRI' THEN '5-FRIDAY' WHEN 'SAT'
THEN '6-SATURDAY'
        WHEN 'SUN' THEN '7-SUNDAY'
    END AS DaySortKey
FROM RESERVATION_FACT rf
JOIN DATE_DIM dd ON rf.reserve_start_date_key = dd.date_key
```

Sample Assignment Report

```
JOIN STAFF_DIM sd ON rf.staff_key = sd.staff_key
/

--
==========================================================================
=======
-- REPORT SECTION 1: DAILY LOAN AND RESERVATION ANALYSIS
--
==========================================================================
=======
TTITLE CENTER
'-------------------------------------------------------------------' SKIP
1 -
CENTER ' PEAK LOAN AND RESERVATION DATE AND STAFFING OPTIMIZATION: ' SKIP 1
-
CENTER ' DAILY LOAN AND RESERVATION ANALYSIS ' SKIP 1 -
CENTER ' YEARS: &P_START_YEAR - &P_END_YEAR | MONTH: &P_MONTH ' SKIP 1 -
CENTER
'-------------------------------------------------------------------' SKIP
1 -
LEFT 'DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN WeekOfMonth              HEADING 'Week'                      FORMAT
A15
COLUMN DayOfMonth               HEADING '     Day of Month'
FORMAT 99
COLUMN TotalActivities          HEADING '    Total Activities'
FORMAT 999,990
COLUMN TotalLoans               HEADING '    Total Loans'
FORMAT 999,990
COLUMN TotalReservations        HEADING '     Total Reservations'
FORMAT 999,990
COLUMN PctLoans                 HEADING '     Loans (%)'
FORMAT 990.00
COLUMN PctReservations          HEADING '     Reservations (%)'
FORMAT 990.00

BREAK ON WeekOfMonth SKIP 1 ON REPORT
COMPUTE SUM LABEL 'Weekly Total' OF TotalActivities TotalLoans
TotalReservations ON WeekOfMonth
COMPUTE SUM LABEL 'Grand Total' OF TotalActivities TotalLoans
TotalReservations ON REPORT

WITH DailyData AS (
    SELECT
        CEIL(day_num_month / 7) as WeekNumber,
        day_num_month,
        COUNT(ActivityID) AS TotalActivities,
        SUM(CASE WHEN ActivityType = 'Loan' THEN 1 ELSE 0 END) AS
```

Sample Assignment Report

```
TotalLoans,
        SUM(CASE WHEN ActivityType = 'Reservation' THEN 1 ELSE 0 END) AS
TotalReservations
    FROM V_ACTIVITY_ANALYSIS
    WHERE cal_year BETWEEN &P_START_YEAR AND &P_END_YEAR
      AND (UPPER(TRIM(cal_month_name)) = UPPER('&P_MONTH') OR
UPPER('&P_MONTH') = 'ALL')
    GROUP BY CEIL(day_num_month / 7), day_num_month
)
SELECT
    'Week ' || WeekNumber AS WeekOfMonth,
    day_num_month AS DayOfMonth,
    TotalActivities,
    TotalLoans,
    TotalReservations,
    (TotalLoans * 100) / NULLIF(SUM(TotalActivities) OVER (), 0) AS
PctLoans,
    (TotalReservations * 100) / NULLIF(SUM(TotalActivities) OVER (), 0) AS
PctReservations
FROM DailyData
ORDER BY day_num_month;

CLEAR BREAKS;
CLEAR COMPUTES;

--
==========================================================================
=======
-- REPORT SECTION 2: DAILY ACTIVITY WORKLOAD DISTRIBUTION
--
==========================================================================
=======
PROMPT
PROMPT

TTITLE CENTER
'-------------------------------------------------------------------' SKIP
1 -
CENTER 'PEAK LOAN AND RESERVATION DATE AND STAFFING OPTIMIZATION:' SKIP 1 -
CENTER 'DAILY ACTIVITY WORKLOAD DISTRIBUTION' SKIP 1 -
CENTER 'YEARS: &P_START_YEAR - &P_END_YEAR | MONTH: &P_MONTH' SKIP 1 -
CENTER
'-------------------------------------------------------------------' SKIP
1 -
LEFT 'DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN day_week                 HEADING 'Day of Week'
FORMAT A25
COLUMN TotalActivities          HEADING 'Total Activities             '
```

Sample Assignment Report

```
FORMAT 999,999
COLUMN AvgActivitiesPerDay      HEADING 'Average Activities per Day   '
FORMAT 99,990.0
COLUMN PctWeeklyWorkload        HEADING 'Share of Workload (%)'
FORMAT A29
COLUMN StaffingFocus            HEADING 'Staffing Focus'
FORMAT A14
COLUMN DaySortKey               NOPRINT

BREAK ON REPORT
COMPUTE SUM LABEL 'Grand Total' OF TotalActivities ON REPORT

WITH DailyActivity AS (
    SELECT
        day_week,
        DaySortKey,
        COUNT(ActivityID) AS TotalActivities,
        COUNT(DISTINCT cal_date) AS NumOfDays
    FROM V_ACTIVITY_ANALYSIS
    WHERE cal_year BETWEEN &P_START_YEAR AND &P_END_YEAR
      AND (UPPER(TRIM(cal_month_name)) = UPPER('&P_MONTH') OR
UPPER('&P_MONTH') = 'ALL')
    GROUP BY day_week, DaySortKey
),
TotalActivity AS (
    SELECT SUM(TotalActivities) AS GrandTotal FROM DailyActivity
)
SELECT
    da.day_week,
    da.TotalActivities,
    da.TotalActivities / da.NumOfDays AS AvgActivitiesPerDay,
    TO_CHAR((da.TotalActivities * 100 / ta.GrandTotal), '990.00') || '%' AS
PctWeeklyWorkload,
    CASE
        WHEN (da.TotalActivities * 100 / ta.GrandTotal) >= 20 THEN 'Peak'
        WHEN (da.TotalActivities * 100 / ta.GrandTotal) >= 15 THEN 'High'
        WHEN (da.TotalActivities * 100 / ta.GrandTotal) >= 10 THEN
'Moderate'
        ELSE 'Low'
    END AS StaffingFocus,
    da.DaySortKey
FROM DailyActivity da, TotalActivity ta
ORDER BY da.DaySortKey;

--
==========================================================================
=======
-- Final Cleanup
--
```

Sample Assignment Report

```
==============================================================================
=======
SPOOL OFF;
TTITLE OFF;
CLEAR COLUMNS;
CLEAR BREAKS;
CLEAR COMPUTES;
SET TERMOUT ON;
SET FEEDBACK ON;
SET VERIFY ON;
UNDEFINE P_START_YEAR;
UNDEFINE P_END_YEAR;
UNDEFINE P_MONTH;
```

Sample Assignment Report

Output:

```
Enter Start Analysis Year (e.g.,       2020): 2020
Enter End Analysis Year (e.g.,       2024): 2024
Enter Month Name (e.g., JANUARY, or ALL for all months): MAY

                                  ------------------------------------------------------------------
                                        PEAK LOAN AND RESERVATION DATE AND STAFFING OPTIMIZATION:
                                                   DAILY LOAN AND RESERVATION ANALYSIS
                                                YEARS:       2020 -       2024 | MONTH: MAY
                                  ------------------------------------------------------------------
DATE: 18-SEP-25                                                                                                                PAGE:    1
```

Week	Day of Month	Total Activities	Total Loans	Total Reservations	Loans (%)	Reservations (%)
Week 1	1	259	150	109	1.88	1.37
	2	243	137	106	1.72	1.33
	3	255	120	135	1.51	1.69
	4	301	155	146	1.95	1.83
	5	215	111	104	1.39	1.31
	6	248	139	109	1.75	1.37
	7	264	156	108	1.96	1.36
**********		---	---	---		
Weekly Total		1,785	968	817		
Week 2	8	236	125	111	1.57	1.39
	9	263	152	111	1.91	1.39
	10	260	136	124	1.71	1.56
	11	246	120	126	1.51	1.58
	12	257	126	131	1.58	1.64
	13	266	146	120	1.83	1.51
	14	250	133	117	1.67	1.47
**********		---	---	---		
Weekly Total		1,778	938	840		
Week 3	15	256	137	119	1.72	1.49
	16	286	150	136	1.88	1.71
	17	254	132	122	1.66	1.53
	18	282	145	137	1.82	1.72
	19	271	147	124	1.85	1.56

Sample Assignment Report

PEAK LOAN AND RESERVATION DATE AND STAFFING OPTIMIZATION:
DAILY LOAN AND RESERVATION ANALYSIS
YEARS: 2020 - 2024 | MONTH: MAY

DATE: 18-SEP-25 PAGE: 2

Week	Day of Month	Total Activities	Total Loans	Total Reservations	Loans (%)	Reservations (%)
Week 3	20	256	141	115	1.77	1.44
	21	256	132	124	1.66	1.56
**********						
Weekly Total		1,861	984	877		
Week 4	22	281	152	129	1.91	1.62
	23	262	134	128	1.68	1.61
	24	248	125	123	1.57	1.54
	25	249	121	128	1.52	1.61
	26	226	118	108	1.48	1.36
	27	276	142	134	1.78	1.68
	28	251	144	107	1.81	1.34
**********						
Weekly Total		1,793	936	857		
Week 5	29	255	149	106	1.87	1.33
	30	245	130	115	1.63	1.44
	31	248	138	110	1.73	1.38
**********						
Weekly Total		748	417	331		
Grand Total		7,965	4,243	3,722		

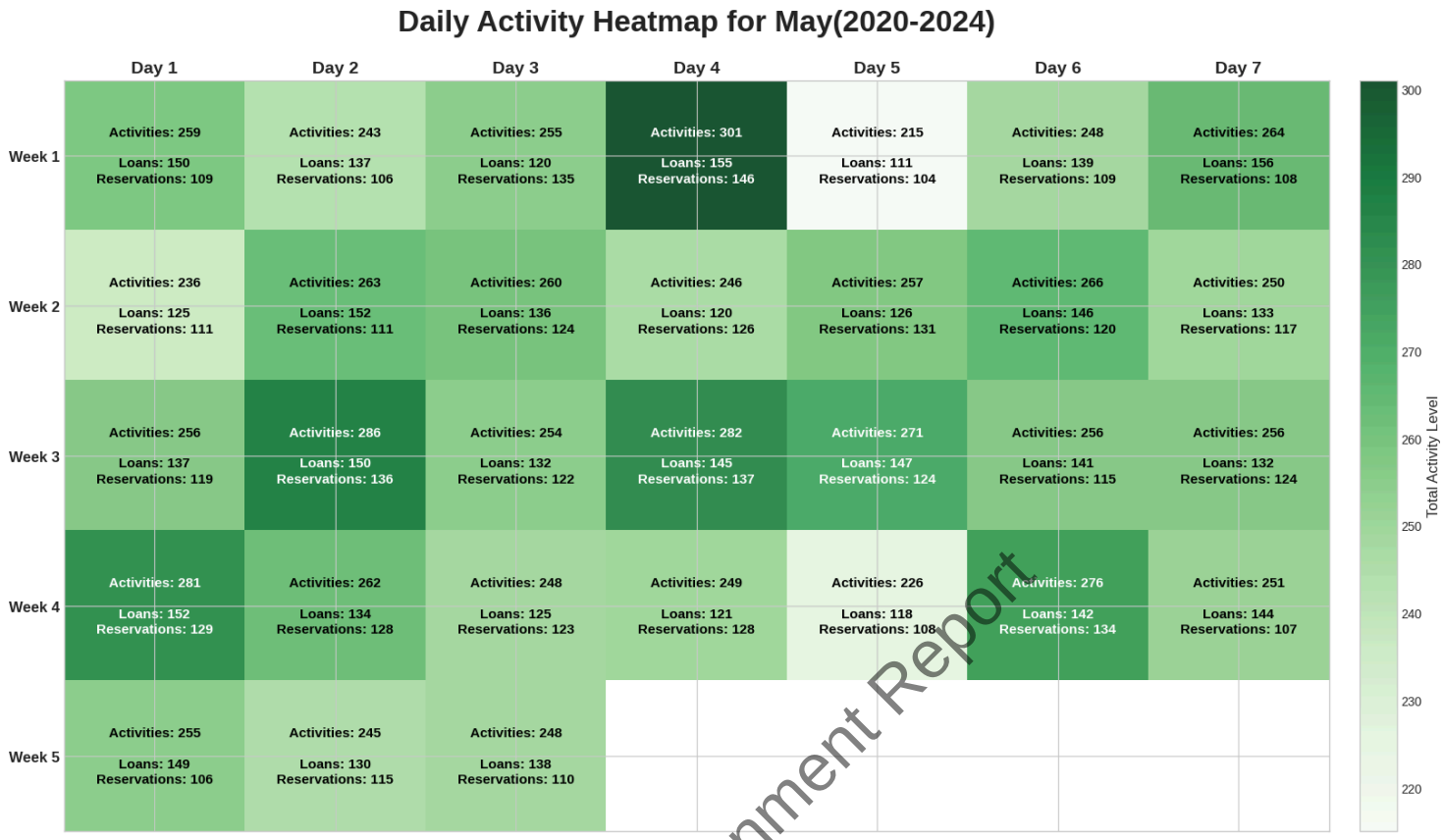
Sample Assignment Report

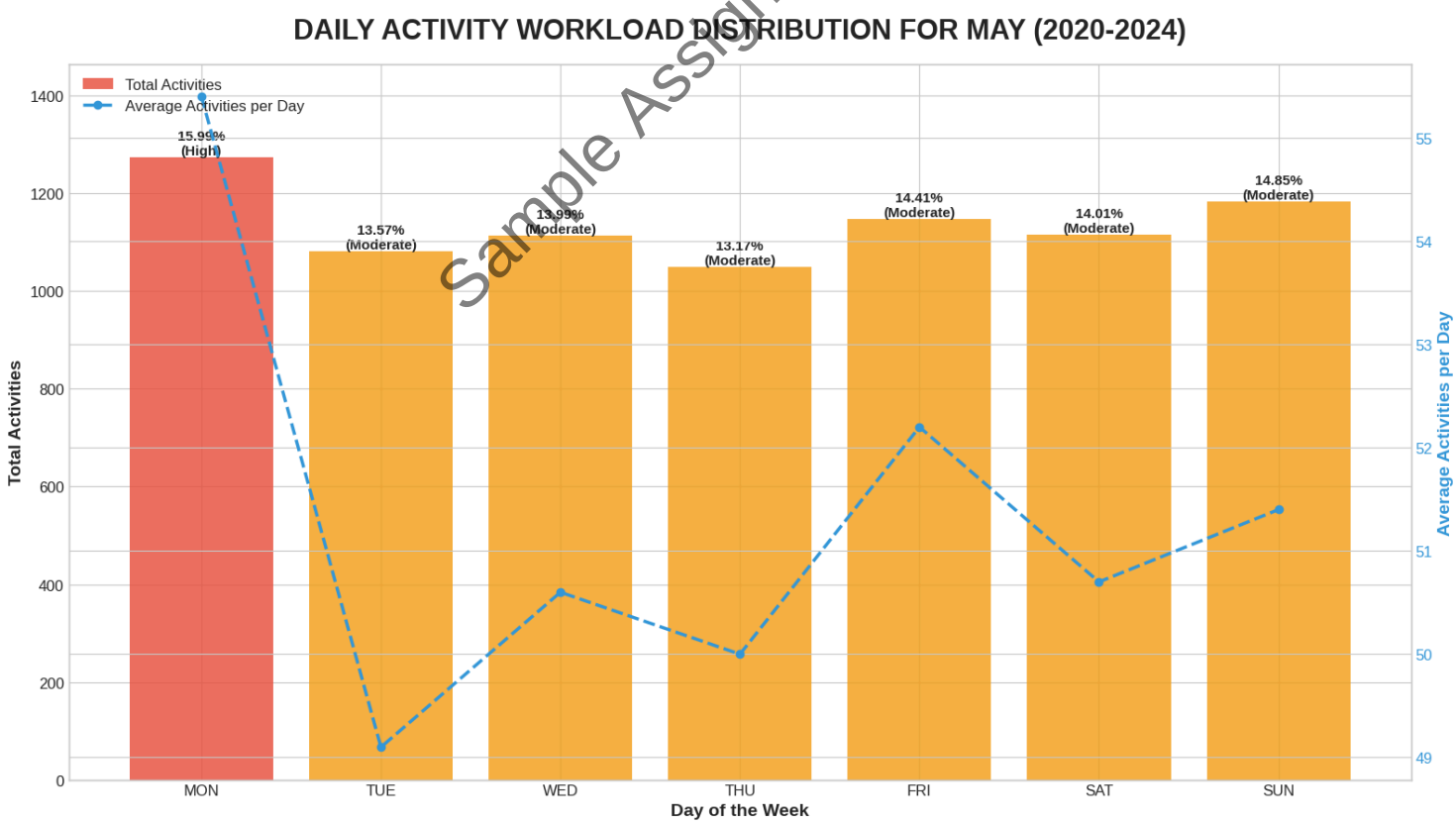
PEAK LOAN AND RESERVATION DATE AND STAFFING OPTIMIZATION:
DAILY ACTIVITY WORKLOAD DISTRIBUTION
YEARS: 2020 - 2024 | MONTH: MAY

DATE: 18-SEP-25 PAGE: 1

Day of Week	Total Activities	Average Activities per Day	Share of Workload (%)	Staffing Focus
MON	1,274	55.4	15.99%	High
TUE	1,081	49.1	13.57%	Moderate
WED	1,114	50.6	13.99%	Moderate
THU	1,049	50.0	13.17%	Moderate
FRI	1,148	52.2	14.41%	Moderate
SAT	1,116	50.7	14.01%	Moderate
SUN	1,183	51.4	14.85%	Moderate
Grand Total	7,965			

Visualisation:





Sample Assignment Report

Justification:

This report focuses on peak loan and reservation and staffing optimisation. Its purpose is to identify peak activity periods to enable data-driven decisions for optimising staff schedules and improving member services. It is made up of two sections, analyzing combined data for the month of May from 2020 to 2024. The first section shows a detailed daily breakdown of monthly activity, while the second looks at the workload distribution for each day of the week.

Daily Loan and Reservation Analysis

Looking at the whole month, it's clear the library isn't equally busy all the time; instead, it has specific, predictable busy periods. The data shows that Week 3 (May 15-21) is the busiest time of the month, with a total of 1,861 loans and reservations. We also see that certain dates are always busy, like May 4th (301 activities) and May 16th (286 activities). Because this mid-month rush happens every single year, we can make a good assumption, this busy period is likely caused by study week and exam week at nearby schools and colleges. This assumption can be proven by the study_week_ind in Date Dimension shows that May 19th to 25th is study week, where students will most likely be busy studying and loaning different types of books as revision material. Capturing this pattern helps the library plan ahead instead of just reacting to a rush.

1. The manager can now schedule an extra part-time helper just for Week 3.
2. Knowing that May 16th is an extra-busy day, they can also decide that no staff can take that day off, and that everyone should focus on helping members, saving paperwork for a quieter time.

Daily Activity and Workload Distribution

Looking at the second section, we can analyze the workload distribution by day. To do this, we use a metric called Share of Workload (%), which is calculated by dividing the total number of activities (loans and reservations) for a specific day of the week by the grand total of all activities for the selected period, then multiplying by 100. Its main purpose is to show the proportion of the total workload that occurs on each day of the week, helping management identify the busiest days and optimize staff schedules accordingly. Staffing focus will change from 'Moderate', to 'High', and to 'Peak' according to Share of Workload when it is more than 10%, 15%, and 20%

Sample Assignment Report

Using this metric, it is clear that Monday is our busiest day of the week in May. It gets almost 16% of all library activity warranting a 'High' staffing focus, which means we need more staff on hand. While workload on the other six days of the week is remarkably stable and falls within the 'Moderate' range. This is likely because people return books from the weekend and check out new ones for the week ahead. On another hand, it may also be caused by an operational bottleneck where our current system is slow or facing staff shortage. To handle this, there are few possible actions we can take.

1. Manager should schedule three staff members at the front desk on Mondays, instead of the usual two, to prevent long lines and keep members happy
2. Tuesday mornings can be set aside for team meetings, and Wednesday can be used for big projects like checking our book stock or adding new books to the system

By scheduling these jobs on quieter days, we make sure our staff are free to help members during the Monday rush, which makes the library run more smoothly for everyone.

3.3.3 Staff Performance Dashboard

SQL Code:

```
--
-------------------------------------------------------------------------------
-------
-- SQL*Plus Environment Setup
--
-------------------------------------------------------------------------------
-------
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
SET PAGESIZE 35
SET LINESIZE 130
SET FEEDBACK OFF
SET VERIFY OFF
SET TERMOUT ON


SPOOL "C:\DW assignment\R3_Enhanced_result.txt"

--
-------------------------------------------------------------------------------
-------
-- User Input Variables
--
-------------------------------------------------------------------------------
-------
COLUMN end_year_col NEW_VALUE P_END_YEAR_DEFAULT
COLUMN start_year_col NEW_VALUE P_START_YEAR_DEFAULT

SELECT
    TO_CHAR(SYSDATE, 'YYYY') - 1 AS end_year_col,
    TO_CHAR(SYSDATE, 'YYYY') - 3 AS start_year_col
FROM DUAL;

ACCEPT P_START_YEAR NUMBER DEFAULT &P_START_YEAR_DEFAULT PROMPT 'Enter Start
Analysis Year (e.g., &P_START_YEAR_DEFAULT): '
ACCEPT P_END_YEAR NUMBER DEFAULT &P_END_YEAR_DEFAULT PROMPT 'Enter End
Analysis Year (e.g., &P_END_YEAR_DEFAULT): '

--
===============================================================================
=======
-- VIEW CREATION: V_STAFF_PERFORMANCE
--
===============================================================================
```

Sample Assignment Report

```
=======
CREATE OR REPLACE VIEW V_STAFF_PERFORMANCE AS
WITH StaffActivities AS (
    SELECT
        dd.cal_year,
        sd.staff_key,
        sd.staffName,
        sd.department,
        sd.position,
        COUNT(lf.loanID) AS total_interactions
    FROM LOAN_FACT lf
    JOIN STAFF_DIM sd ON lf.staff_key = sd.staff_key
    JOIN DATE_DIM dd ON lf.date_key = dd.date_key
    GROUP BY dd.cal_year, sd.staff_key, sd.staffName, sd.department,
sd.position
    UNION ALL
    SELECT
        dd.cal_year,
        sd.staff_key,
        sd.staffName,
        sd.department,
        sd.position,
        COUNT(rf.reserveID) AS total_interactions
    FROM RESERVATION_FACT rf
    JOIN STAFF_DIM sd ON rf.staff_key = sd.staff_key
    JOIN DATE_DIM dd ON rf.reserve_start_date_key = dd.date_key
    GROUP BY dd.cal_year, sd.staff_key, sd.staffName, sd.department,
sd.position
),
StaffSales AS (
    SELECT
        dd.cal_year,
        sd.staff_key,
        sd.staffName,
        sd.department,
        sd.position,
        SUM(sf.orderTotalprice) AS total_sales
    FROM SALES_FACT sf
    JOIN STAFF_DIM sd ON sf.staff_key = sd.staff_key
    JOIN DATE_DIM dd ON sf.date_key = dd.date_key
    GROUP BY dd.cal_year, sd.staff_key, sd.staffName, sd.department,
sd.position
)
SELECT
    COALESCE(sa.cal_year, ss.cal_year) AS cal_year,
    COALESCE(sa.staff_key, ss.staff_key) AS staff_key,
    COALESCE(sa.staffName, ss.staffName) AS staffName,
    COALESCE(sa.department, ss.department) AS department,
    COALESCE(sa.position, ss.position) AS position,
```

Sample Assignment Report

```
    SUM(COALESCE(sa.total_interactions, 0)) AS TotalInteractions,
    SUM(COALESCE(ss.total_sales, 0)) AS TotalSales
FROM StaffActivities sa
FULL OUTER JOIN StaffSales ss ON sa.staff_key = ss.staff_key AND sa.cal_year
= ss.cal_year
GROUP BY COALESCE(sa.cal_year, ss.cal_year), COALESCE(sa.staff_key,
ss.staff_key), COALESCE(sa.staffName, ss.staffName), COALESCE(sa.department,
ss.department), COALESCE(sa.position, ss.position)
/

--
==========================================================================
=======
-- REPORT SECTION 1: STAFF PERFORMANCE LEADERBOARD
--
==========================================================================
=======
TTITLE CENTER '-------------------------------------------------------------' SKIP
1 -
CENTER 'STAFF PERFORMANCE DASHBOARD:' SKIP 1 -
CENTER 'PERFORMANCE LEADERBOARD' SKIP 1 -
CENTER 'YEARS: &P_START_YEAR - &P_END_YEAR' SKIP 1 -
CENTER '-------------------------------------------------------------' SKIP 1 -
LEFT 'DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN department               HEADING 'Department'            FORMAT A20
COLUMN position                 HEADING 'Position'              FORMAT A20
COLUMN staffName                HEADING 'Staff Name'            FORMAT A15
COLUMN TotalSales               HEADING '      Total Sales(RM)' FORMAT
9,999,990.00
COLUMN TotalInteractions        HEADING ' Total Interactions'   FORMAT
99,990
COLUMN SalesRank                HEADING ' Sales Rank'           FORMAT 990
COLUMN InteractionRank          HEADING '  Interaction Rank'    FORMAT 990

BREAK ON department SKIP 1 ON position SKIP 1

WITH StaffTotals AS (
    SELECT
        department,
        position,
        staffName,
        SUM(TotalSales) AS TotalSales,
        SUM(TotalInteractions) AS TotalInteractions
    FROM V_STAFF_PERFORMANCE
    WHERE cal_year BETWEEN &P_START_YEAR AND &P_END_YEAR
      AND department IN ('OPERATIONS', 'SALES')
    GROUP BY department, position, staffName
)
```

Sample Assignment Report

```
SELECT
    department,
    position,
    staffName,
    TotalSales,
    TotalInteractions,
    RANK() OVER (ORDER BY TotalSales DESC) AS SalesRank,
    RANK() OVER (ORDER BY TotalInteractions DESC) AS InteractionRank
FROM StaffTotals
ORDER BY department, position, SalesRank, InteractionRank;

CLEAR BREAKS;
CLEAR COMPUTES;

--
==========================================================================
=======
-- REPORT SECTION 2: MULTI-YEAR PERFORMANCE TREND (WITH GROWTH RATE)
--
==========================================================================
=======
PROMPT
PROMPT

TTITLE CENTER '-------------------------------------------------------' SKIP
1 -
CENTER 'STAFF PERFORMANCE DASHBOARD:' SKIP 1 -
CENTER 'MULTI-YEAR PERFORMANCE TREND' SKIP 1 -
CENTER 'YEARS: &P_START_YEAR - &P_END_YEAR' SKIP 1 -
CENTER '-------------------------------------------------------' SKIP 1 -
LEFT 'DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

-- Updated Column Formatting for Section 2
COLUMN department               HEADING 'Department'                FORMAT
A15
COLUMN position                 HEADING 'Position'                  FORMAT
A19
COLUMN staffName                HEADING 'Staff Name'                FORMAT
A19
COLUMN cal_year                 HEADING 'Year'                      FORMAT
9999
COLUMN TotalSales               HEADING '   Sales(RM)'              FORMAT
999,990.00
COLUMN SalesGrowthPct           HEADING ' Sales Growth(%)'          FORMAT
A16
COLUMN TotalInteractions        HEADING '   Interactions'           FORMAT
99,990
COLUMN InteractionGrowthPct     HEADING ' Interaction Growth(%)'    FORMAT
A22
```

Sample Assignment Report

```
BREAK ON department SKIP 1 ON position SKIP 1 ON staffName SKIP 1

WITH YearlyPerformance AS (
    SELECT
        staff_key,
        department,
        position,
        staffName,
        cal_year,
        TotalSales,
        TotalInteractions,
        LAG(TotalSales, 1, 0) OVER (PARTITION BY staff_key ORDER BY
cal_year) AS SalesPY,
        LAG(TotalInteractions, 1, 0) OVER (PARTITION BY staff_key ORDER BY
cal_year) AS InteractionsPY
    FROM V_STAFF_PERFORMANCE
    WHERE cal_year BETWEEN &P_START_YEAR - 1 AND &P_END_YEAR -- Fetch one
prior year for growth calculation
      AND department IN ('OPERATIONS', 'SALES')
)
SELECT
    department,
    position,
    staffName,
    cal_year,
    TotalSales,
    CASE
        WHEN SalesPY = 0 THEN 'N/A'
        ELSE TO_CHAR(((TotalSales - SalesPY) / SalesPY) * 100, '990.00') ||
'%'
    END AS SalesGrowthPct,
    TotalInteractions,
    CASE
        WHEN InteractionsPY = 0 THEN 'N/A'
        ELSE TO_CHAR(((TotalInteractions - InteractionsPY) / InteractionsPY)
* 100, '990.00') || '%'
    END AS InteractionGrowthPct
FROM YearlyPerformance
WHERE cal_year BETWEEN &P_START_YEAR AND &P_END_YEAR -- Filter out the extra
previous year from final display
ORDER BY department, position, staffName, cal_year;


CLEAR BREAKS;
CLEAR COMPUTES;

--
==============================================================================
```

Sample Assignment Report

```
=======
-- REPORT SECTION 3: STRATEGIC PERFORMANCE QUADRANTS (Aggregated)
--
==============================================================================
=======
PROMPT
PROMPT

TTITLE CENTER '-------------------------------------------------------' SKIP
1 -
CENTER 'STAFF PERFORMANCE DASHBOARD:' SKIP 1 -
CENTER 'STRATEGIC PERFORMANCE QUADRANTS' SKIP 1 -
CENTER 'YEARS: &P_START_YEAR - &P_END_YEAR' SKIP 1 -
CENTER '-------------------------------------------------------' SKIP 1 -
LEFT 'DATE: ' _DATE RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN PerformanceQuadrant      HEADING 'Performance Quadrant'  FORMAT A27
COLUMN department               HEADING 'Department'            FORMAT A17
COLUMN position                 HEADING 'Position'              FORMAT A20
COLUMN staffName                HEADING 'Staff Name'            FORMAT A20
COLUMN TotalSales               HEADING '     Total Sales(RM)'  FORMAT
9,999,990.00
COLUMN TotalInteractions        HEADING '   Total Interactions' FORMAT
99,990

BREAK ON PerformanceQuadrant SKIP 1 ON department SKIP 1 ON position SKIP 1

WITH StaffTotals AS (
    SELECT
        staffName,
        department,
        position,
        SUM(TotalSales) AS TotalSales,
        SUM(TotalInteractions) AS TotalInteractions
    FROM V_STAFF_PERFORMANCE
    WHERE cal_year BETWEEN &P_START_YEAR AND &P_END_YEAR
      AND department IN ('OPERATIONS', 'SALES')
    GROUP BY staffName, department, position
),
StaffAverages AS (
    SELECT
        staffName,
        department,
        position,
        TotalSales,
        TotalInteractions,
        AVG(CASE WHEN department = 'SALES' THEN TotalSales END) OVER () AS
AvgSales,
        AVG(CASE WHEN department = 'OPERATIONS' THEN TotalInteractions END)
```

Sample Assignment Report

```
OVER () AS AvgInteractions
    FROM StaffTotals
)
SELECT
    CASE
        WHEN TotalSales >= AvgSales AND TotalInteractions >= AvgInteractions
THEN 'EMPTY'
        WHEN department = 'SALES' AND TotalSales >= AvgSales THEN '1. Sales
Specialists'
        WHEN department = 'OPERATIONS' AND TotalInteractions >=
AvgInteractions THEN '2. Engagement Champions'
        ELSE '3. Growth Potential'
    END AS PerformanceQuadrant,
    department,
    position,
    staffName,
    TotalSales,
    TotalInteractions
FROM StaffAverages
ORDER BY PerformanceQuadrant, department, position, TotalSales DESC;

--
==========================================================================
=======
-- Final Cleanup
--
==========================================================================
=======
SPOOL OFF;
TTITLE OFF;
CLEAR COLUMNS;
CLEAR BREAKS;
CLEAR COMPUTES;
SET TERMOUT ON;
SET FEEDBACK ON;
SET VERIFY ON;
UNDEFINE P_START_YEAR;
UNDEFINE P_END_YEAR;
```

Sample Assignment Report

Output:

```
Enter Start Analysis Year (e.g.,       2022): 2022
Enter End Analysis Year (e.g.,       2024): 2024

                                    -------------------------------------------------------
                                                 STAFF PERFORMANCE DASHBOARD:
                                                    PERFORMANCE LEADERBOARD
                                               YEARS:       2022 -       2024
                                    -------------------------------------------------------
DATE: 18-SEP-25                                                                                                                    PAGE:    1

Department           Position             Staff Name            Total Sales(RM)  Total Interactions  Sales Rank   Interaction Rank
-------------------- -------------------- --------------- --------------------- ------------------- ----------- ------------------
OPERATIONS           ASSISTANT MANAGER    GOH WEI                        0.00               4,509           7                  8

                     MANAGER              CHONG MEI                      0.00               9,256           7                  1

                     OFFICER              CHONG LI                       0.00               6,531           7                  4
                                          CHONG HUA                      0.00               4,575           7                  5
                                          TAN BO                         0.00               4,511           7                  7
                                          LAU MING                       0.00               4,481           7                  9

                     SENIOR OFFICER       LIM MING                       0.00               9,201           7                  2
                                          CHONG MING                     0.00               9,115           7                  3
                                          LEE XIU                        0.00               4,525           7                  6


SALES                CLERKS               WONG MEI               1,287,939.45                   0           4                 10
                                          CHEONG MING            1,286,484.71                   0           5                 10

                     MANAGER              LIM JIN                1,911,003.38                   0           2                 10

                     OFFICER              LIM TIAN               1,945,834.69                   0           1                 10
                                          LAU MEI                  657,319.00                   0           6                 10

                     SENIOR OFFICER       GOH HUA                1,301,334.21                   0           3                 10
```

Sample Assignment Report

STAFF PERFORMANCE DASHBOARD:
MULTI-YEAR PERFORMANCE TREND
YEARS: 2022 - 2024

DATE: 18-SEP-25 PAGE: 1

Department	Position	Staff Name	Year	Sales(RM)	Sales Growth(%)	Interactions	Interaction Growth(%)
OPERATIONS	ASSISTANT MANAGER	GOH WEI	2022	0.00	N/A	1,503	1.69%
			2023	0.00	N/A	1,501	-0.13%
			2024	0.00	N/A	1,505	0.27%
	MANAGER	CHONG MEI	2022	0.00	N/A	3,042	0.93%
			2023	0.00	N/A	3,110	2.24%
			2024	0.00	N/A	3,104	-0.19%
	OFFICER	CHONG HUA	2022	0.00	N/A	1,500	-2.60%
			2023	0.00	N/A	1,534	2.27%
			2024	0.00	N/A	1,541	0.46%
		CHONG LI	2022	0.00	N/A	2,164	-3.18%
			2023	0.00	N/A	2,204	1.85%
			2024	0.00	N/A	2,163	-1.86%
		LAU MING	2022	0.00	N/A	1,470	-0.81%
			2023	0.00	N/A	1,487	1.16%
			2024	0.00	N/A	1,524	2.49%
		TAN BO	2022	0.00	N/A	1,498	-0.60%
			2023	0.00	N/A	1,510	0.80%
			2024	0.00	N/A	1,503	-0.46%

STAFF PERFORMANCE DASHBOARD:
MULTI-YEAR PERFORMANCE TREND
YEARS: 2022 - 2024

DATE: 18-SEP-25 PAGE: 2

Department	Position	Staff Name	Year	Sales(RM)	Sales Growth(%)	Interactions	Interaction Growth(%)
OPERATIONS	SENIOR OFFICER	CHONG MING	2022	0.00	N/A	3,001	-4.67%
			2023	0.00	N/A	3,020	0.63%
			2024	0.00	N/A	3,094	2.45%
		LEE XIU	2022	0.00	N/A	1,499	-3.29%
			2023	0.00	N/A	1,529	2.00%
			2024	0.00	N/A	1,497	-2.09%
		LIM MING	2022	0.00	N/A	3,055	1.77%
			2023	0.00	N/A	3,086	1.01%
			2024	0.00	N/A	3,060	-0.84%
SALES	CLERKS	CHEONG MING	2022	422,970.98	-3.10%	0	N/A
			2023	430,383.88	1.75%	0	N/A
			2024	433,079.85	0.63%	0	N/A
		WONG MEI	2022	413,364.39	-3.33%	0	N/A
			2023	432,199.80	4.56%	0	N/A
			2024	442,375.26	2.35%	0	N/A

Sample Assignment Report

STAFF PERFORMANCE DASHBOARD:
MULTI-YEAR PERFORMANCE TREND
YEARS: 2022 - 2024

DATE: 18-SEP-25 PAGE: 3

Department	Position	Staff Name	Year	Sales(RM)	Sales Growth(%)	Interactions	Interaction Growth(%)
SALES	MANAGER	LIM JIN	2022	645,787.25	-2.22%	0	N/A
			2023	633,448.83	-1.91%	0	N/A
			2024	631,767.30	-0.27%	0	N/A
	OFFICER	LAU MEI	2022	220,473.44	1.21%	0	N/A
			2023	218,088.25	-1.08%	0	N/A
			2024	218,757.31	0.31%	0	N/A
		LIM TIAN	2022	632,053.63	-2.73%	0	N/A
			2023	633,102.13	0.17%	0	N/A
			2024	680,678.93	7.51%	0	N/A
	SENIOR OFFICER	GOH HUA	2022	440,742.62	-1.38%	0	N/A
			2023	430,153.43	-2.40%	0	N/A
			2024	430,438.16	0.07%	0	N/A

Sample Assignment Report

STAFF PERFORMANCE DASHBOARD:
STRATEGIC PERFORMANCE QUADRANTS
YEARS: 2022 - 2024

DATE: 18-SEP-25 PAGE: 1

Performance Quadrant	Department	Position	Staff Name	Total Sales(RM)	Total Interactions
1. Sales Specialists	SALES	MANAGER	LIM JIN	1,911,003.38	0
		OFFICER	LIM TIAN	1,945,834.69	0
2. Engagement Champions	OPERATIONS	MANAGER	CHONG MEI	0.00	9,256
		OFFICER	CHONG LI	0.00	6,531
		SENIOR OFFICER	CHONG MING	0.00	9,115
			LIM MING	0.00	9,201
3. Growth Potential	OPERATIONS	ASSISTANT MANAGER	GOH WEI	0.00	4,509
		OFFICER	CHONG HUA	0.00	4,575
			TAN BO	0.00	4,511
			LAU MING	0.00	4,481
		SENIOR OFFICER	LEE XIU	0.00	4,525
	SALES	CLERKS	WONG MEI	1,287,939.45	0

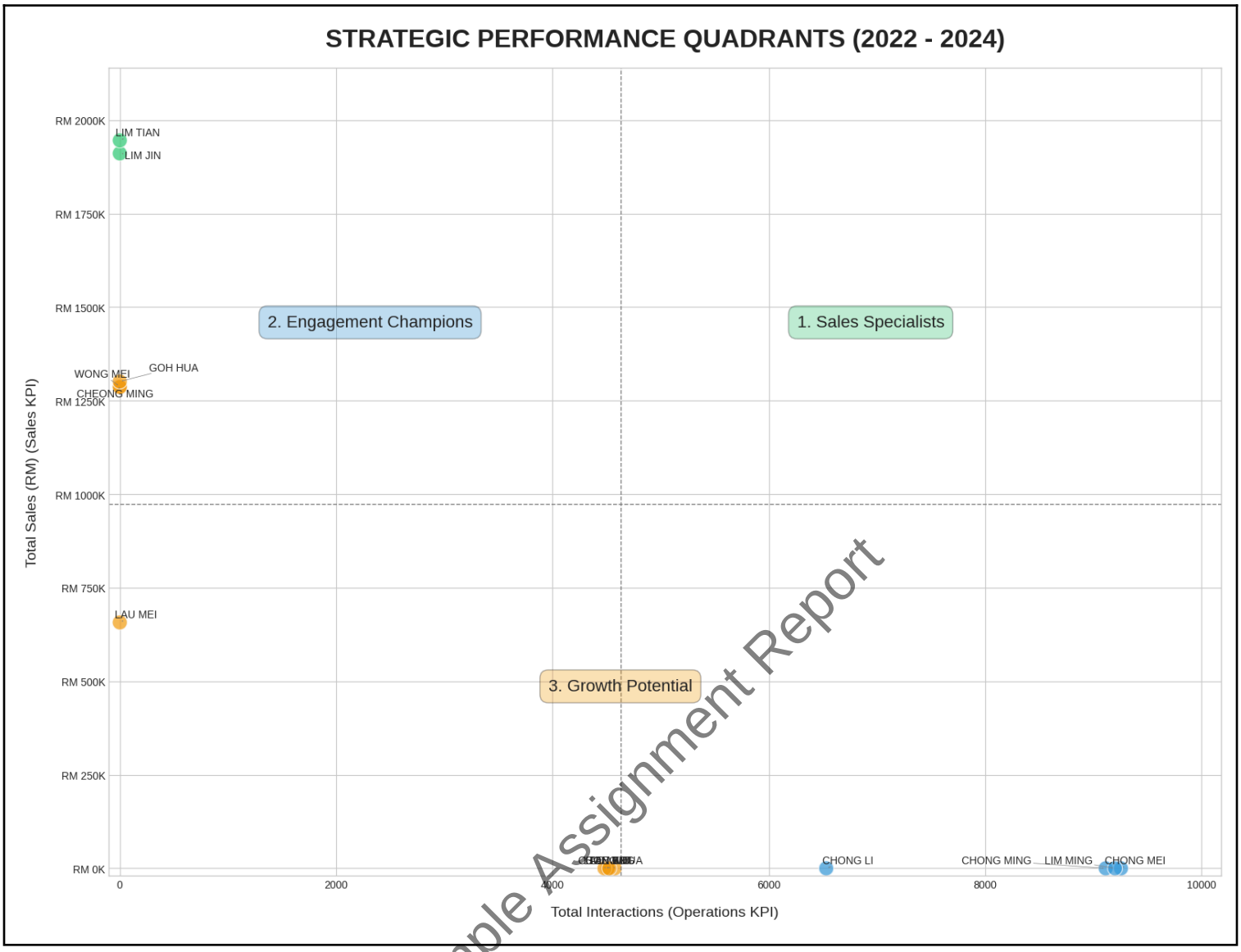
STAFF PERFORMANCE DASHBOARD:
STRATEGIC PERFORMANCE QUADRANTS
YEARS: 2022 - 2024

DATE: 18-SEP-25 PAGE: 2

Performance Quadrant	Department	Position	Staff Name	Total Sales(RM)	Total Interactions
3. Growth Potential	SALES	CLERKS	CHEONG MING	1,286,434.71	0
		OFFICER	LAU MEI	657,319.00	0
		SENIOR OFFICER	GOH HUA	1,301,334.21	0

Sample Assignment Report

Visualisation:



Sample Assignment Report

Justification:

This report, which focuses on the staff performance dashboard from year 2022 to year 2024, is made up of 3 sections. It shows a three-year performance leaderboard to identify our long-term top performers, a multi-year performance trend analysis to evaluate individual growth, and a strategic performance quadrant analysis to guide targeted staff development.

Performance Leaderboard

The first section identifies our most impactful employees based on their total contributions over the past three years. In the Operations department, which is measured by member interactions, a clear group of leaders has emerged. Chong Mei, Lim Ming, and Chong Ming have been our most active staff, each handling over 9,000 interactions. In the Sales department, measured by revenue, Lim Tian and Lim Jin are our standout performers, both generating close to RM 2 million in sales. To retain these key, long-term contributors, it is vital to acknowledge their success.

1. Implement a tiered annual bonus structure where top-tier performers like Lim Tian and Chong Mei, who

hold the #1 ranks, receive a premium reward, publicly celebrating their sustained excellence and setting a clear benchmark for success.

Multi-Year Performance Trend

The second section compares the year 2022 to 2024, revealing the performance of each employee. In the Sales department, the data tells two entire different stories. We see that Lim Tian is not just a top performer, but a rapidly accelerating one, with his sales growth jumping to 7.51% in 2024. Given this sharp increase, it's a strong assumption that Lim Tian has successfully mastered a new sales strategy or is excelling with a recently introduced product line. This kind of outlier growth is rarely accidental and suggests a specific, replicable method.

1. The manager should have a meeting with Lim Tian and turn his personal success into a training guide for the whole team.

On the other hand, our second-highest salesperson Lim Jin, is a big worry because his sales have dropped for three years in a row. Based on this steady drop, we can assume he might be using old ways of selling or talking to customers who aren't buying as much anymore. We need to take helpful action right away.

2. A friendly meeting between Lim Jin and Lim Tian should be hosted. This isn't for criticism but for them to work together to help Lim Jin find new ways to sell, with a clear goal of turning his sales numbers around next quarter.

In the Operations department, we see a worrying pattern where our two best performing staff, Chong Mei and Lim Ming, both had their numbers dip in 2024. It's likely that as senior staff, they were given extra manager-type tasks or paperwork, which took time away from helping members one-on-one. A drop for both leaders at the same time suggests their jobs have changed, not that they are underperforming.

3. A manager should track what they do all week. If it shows they are spending a lot of time on other tasks, their goals should be formally changed to match their new duties.

At the same time, we must notice and reward the strong, steady growth of staff like Lau Ming (2.49% growth) and Chong Ming (2.45% growth). They are clearly finding good ways to work, hence we should

4. Give one of them the chance to lead a small test project on how to better engage with members, using their success to help the whole team.

Strategic Performance Quadrants

The final section helps us plan for the future by sorting staff into performance groups. Our top performers with ability above average are identified as "Sales Specialists" and "Engagement Champions." Most other staff are in the "Growth Potential" category, which is an opportunity for us to provide targeted support. We should use our top performers as mentors for those in the growth category, move beyond generic training to a formal, data-driven mentorship program. By providing specific training and creating partnership programs, we can help all employees improve, which will make our entire library more effective and competitive.

1. Lim Tian, the sales growth champion, should be promoted to a higher position as an approval of his performance, and allow him to lead the department to achieve higher sales.
2. Lau Ming, with 2.49% growth, can be partnered with Lee Xiu, who experienced a 2.09% decline for weekly co-working sessions, allowing Lau Ming to share his successful engagement techniques directly.

Sample Assignment Report

3.4 Wong De Lek

3.4.1 Quarterly Profitability Deep Dive by Book Category and Publisher

SQL Code:

```
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES

SET DEFINE ON
SET ESCAPE ON
SET PAGESIZE 60
SET LINESIZE 180
SET FEEDBACK OFF
SET VERIFY OFF
SET ECHO OFF
SET TERMOUT ON
SET TRIMSPOOL ON

SPOOL quarterly_profitability_output.txt

ACCEPT start_year NUMBER PROMPT 'Enter the start year (e.g., 2016): '
ACCEPT end_year   NUMBER PROMPT 'Enter the end year   (e.g., 2025): '
ACCEPT category   CHAR   PROMPT 'Enter book category (or ALL for all
categories): '

COLUMN run_dt NEW_VALUE run_dt
SELECT TO_CHAR(SYSDATE,'DD-MON-YYYY') AS run_dt FROM dual;

COLUMN start_y NEW_VALUE start_y
SELECT &start_year AS start_y FROM dual;

COLUMN end_y NEW_VALUE end_y
SELECT &end_year AS end_y FROM dual;

COLUMN cat_label NEW_VALUE cat_label
SELECT CASE
         WHEN UPPER(TRIM('&category')) IN ('','ALL') THEN 'ALL CATEGORIES'
         ELSE UPPER(TRIM('&category'))
       END AS cat_label
FROM dual;


TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'QUARTERLY PROFITABILITY DEEP DIVE - &cat_label (&start_y -
&end_y)' SKIP 1 -
```

Sample Assignment Report

```
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN rnk         HEADING 'RANK'          FORMAT 999
COLUMN year        HEADING 'YEAR'          FORMAT 9999
COLUMN quarter     HEADING 'QUARTER'       FORMAT A8
COLUMN category    HEADING 'CATEGORY'      FORMAT A18
COLUMN publisher   HEADING 'PUBLISHER'     FORMAT A30
COLUMN total_rev   HEADING 'TOTAL REVENUE' FORMAT 999,999,999.99
COLUMN cogs_est    HEADING 'ESTIMATED COGS' FORMAT 999,999,999.99
COLUMN gross_prof  HEADING 'GROSS PROFIT'   FORMAT 999,999,999.99
COLUMN margin_pct  HEADING 'MARGIN %'       FORMAT 999.99

BREAK ON REPORT
COMPUTE SUM LABEL 'TOTAL'   OF total_rev cogs_est gross_prof ON REPORT
COMPUTE AVG LABEL 'AVG.' OF margin_pct ON REPORT

WITH cost_avg AS (
  SELECT pf.book_key,
         CASE
           WHEN SUM(pf.purchaseQuantity) = 0 THEN 0
           ELSE SUM(pf.purchaseQuantity * pf.purchaseUnitCost) /
NULLIF(SUM(pf.purchaseQuantity), 0)
         END AS avg_unit_cost
  FROM purchase_fact pf
  GROUP BY pf.book_key
),
profitability AS (
  SELECT
    d.cal_year AS year,
    d.cal_year_quarter AS quarter,
    NVL(b.bookCategory, 'UNKNOWN') AS category,
    NVL(b.publisherName, 'UNKNOWN') AS publisher,
    ROUND(SUM(sf.orderTotalPrice), 2) AS total_rev,
    ROUND(SUM(sf.orderQty * NVL(c.avg_unit_cost, 0)), 2) AS cogs_est,
    ROUND(SUM(sf.orderTotalPrice) - SUM(sf.orderQty * NVL(c.avg_unit_cost,
0)), 2) AS gross_prof,
    CASE
      WHEN SUM(sf.orderTotalPrice) = 0 THEN 0
      ELSE ROUND(((SUM(sf.orderTotalPrice) - SUM(sf.orderQty *
NVL(c.avg_unit_cost, 0))) /
                  NULLIF(SUM(sf.orderTotalPrice), 0)) * 100, 2)
    END AS margin_pct
  FROM sales_fact sf
  JOIN date_dim d ON d.date_key = sf.date_key
  JOIN book_dim b ON b.book_key = sf.book_key
  LEFT JOIN cost_avg c ON c.book_key = sf.book_key
  WHERE d.cal_year BETWEEN &start_year AND &end_year
```

Sample Assignment Report

```
    AND (UPPER(TRIM('&category')) IN ('', 'ALL') OR
         UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP BY d.cal_year, d.cal_year_quarter, b.bookCategory, b.publisherName
)
SELECT
  RANK() OVER (PARTITION BY year, quarter ORDER BY gross_prof DESC) AS rnk,
  year,
  quarter,
  category,
  publisher,
  total_rev,
  cogs_est,
  gross_prof,
  margin_pct
FROM profitability
ORDER BY year, quarter, rnk;

CLEAR BREAKS
CLEAR COMPUTES

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'ANNUAL PROFITABILITY SUMMARY BY CATEGORY (&start_y - &end_y)'
SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN year2       HEADING 'YEAR'           FORMAT 9999
COLUMN category2   HEADING 'CATEGORY'       FORMAT A18
COLUMN total_rev2  HEADING 'TOTAL REVENUE' FORMAT 999,999,999.99
COLUMN cogs_est2   HEADING 'ESTIMATED COGS' FORMAT 999,999,999.99
COLUMN gross_prof2 HEADING 'GROSS PROFIT'   FORMAT 999,999,999.99
COLUMN margin_pct2 HEADING 'MARGIN %'       FORMAT 999.99

BREAK ON REPORT
COMPUTE SUM LABEL 'TOTAL'   OF total_rev2 cogs_est2 gross_prof2 ON REPORT
COMPUTE AVG LABEL 'AVERAGE' OF margin_pct2 ON REPORT

WITH cost_avg AS (
  SELECT pf.book_key,
         CASE
           WHEN SUM(pf.purchaseQuantity) = 0 THEN 0
           ELSE SUM(pf.purchaseQuantity * pf.purchaseUnitCost) /
NULLIF(SUM(pf.purchaseQuantity), 0)
         END AS avg_unit_cost
  FROM purchase_fact pf
  GROUP BY pf.book_key
),
```

Sample Assignment Report

```
annual AS (
  SELECT
    d.cal_year AS year2,
    NVL(b.bookCategory, 'UNKNOWN') AS category2,
    ROUND(SUM(sf.orderTotalPrice), 2) AS total_rev2,
    ROUND(SUM(sf.orderQty * NVL(c.avg_unit_cost, 0)), 2) AS cogs_est2,
    ROUND(SUM(sf.orderTotalPrice) - SUM(sf.orderQty * NVL(c.avg_unit_cost,
0)), 2) AS gross_prof2,
    CASE
      WHEN SUM(sf.orderTotalPrice) = 0 THEN 0
      ELSE ROUND(((SUM(sf.orderTotalPrice) - SUM(sf.orderQty *
NVL(c.avg_unit_cost, 0))) /
                  NULLIF(SUM(sf.orderTotalPrice), 0)) * 100, 2)
    END AS margin_pct2
  FROM sales_fact sf
  JOIN date_dim d ON d.date_key = sf.date_key
  JOIN book_dim b ON b.book_key = sf.book_key
  LEFT JOIN cost_avg c ON c.book_key = sf.book_key
  WHERE d.cal_year BETWEEN &start_year AND &end_year
    AND (UPPER(TRIM('&category')) IN ('', 'ALL') OR
         UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP BY d.cal_year, b.bookCategory
)
SELECT year2, category2, total_rev2, cogs_est2, gross_prof2, margin_pct2
FROM annual
ORDER BY year2, category2;

CLEAR BREAKS
CLEAR COMPUTES

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'TOP 5 MOST PROFITABLE CATEGORIES (&start_y - &end_y)' SKIP 1
-
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN rank_num    HEADING 'RANK'          FORMAT 99
COLUMN cat         HEADING 'CATEGORY'      FORMAT A25
COLUMN cat_profit  HEADING 'TOTAL PROFIT'  FORMAT 999,999,999.99
COLUMN cat_margin  HEADING 'AVG MARGIN %'  FORMAT 999.99

WITH cost_avg AS (
  SELECT pf.book_key,
         CASE
           WHEN SUM(pf.purchaseQuantity) = 0 THEN 0
           ELSE SUM(pf.purchaseQuantity * pf.purchaseUnitCost) /
NULLIF(SUM(pf.purchaseQuantity), 0)
```

Sample Assignment Report

```
         END AS avg_unit_cost
  FROM purchase_fact pf
  GROUP BY pf.book_key
),
category_profit AS (
  SELECT
    NVL(b.bookCategory, 'UNKNOWN') AS cat,
    SUM(sf.orderTotalPrice) - SUM(sf.orderQty * NVL(c.avg_unit_cost, 0)) AS
profit,
    CASE
      WHEN SUM(sf.orderTotalPrice) = 0 THEN 0
      ELSE ((SUM(sf.orderTotalPrice) - SUM(sf.orderQty *
NVL(c.avg_unit_cost, 0))) /
            NULLIF(SUM(sf.orderTotalPrice), 0)) * 100
    END AS margin_pct
  FROM sales_fact sf
  JOIN date_dim d ON d.date_key = sf.date_key
  JOIN book_dim b ON b.book_key = sf.book_key
  LEFT JOIN cost_avg c ON c.book_key = sf.book_key
  WHERE d.cal_year BETWEEN &start_year AND &end_year
    AND (UPPER(TRIM('&category')) IN ('', 'ALL') OR
         UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP BY b.bookCategory
)
SELECT
  ROWNUM AS rank_num,
  cat,
  ROUND(profit, 2) AS cat_profit,
  ROUND(margin_pct, 2) AS cat_margin
FROM (
  SELECT cat, profit, margin_pct
  FROM category_profit
  ORDER BY profit DESC
)
WHERE ROWNUM <= 5;

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'TOP 5 MOST PROFITABLE PUBLISHERS (&start_y - &end_y)' SKIP 1
-
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2

COLUMN rank_num2   HEADING 'RANK'          FORMAT 99
COLUMN pub         HEADING 'PUBLISHER'     FORMAT A35
COLUMN pub_profit  HEADING 'TOTAL PROFIT'  FORMAT 999,999,999.99
COLUMN pub_margin  HEADING 'AVG MARGIN %'  FORMAT 999.99
```

Sample Assignment Report

```
WITH cost_avg AS (
  SELECT pf.book_key,
         CASE
           WHEN SUM(pf.purchaseQuantity) = 0 THEN 0
           ELSE SUM(pf.purchaseQuantity * pf.purchaseUnitCost) /
NULLIF(SUM(pf.purchaseQuantity), 0)
         END AS avg_unit_cost
  FROM purchase_fact pf
  GROUP BY pf.book_key
),
publisher_profit AS (
  SELECT
    NVL(b.publisherName, 'UNKNOWN') AS pub,
    SUM(sf.orderTotalPrice) - SUM(sf.orderQty * NVL(c.avg_unit_cost, 0)) AS
profit,
    CASE
      WHEN SUM(sf.orderTotalPrice) = 0 THEN 0
      ELSE ((SUM(sf.orderTotalPrice) - SUM(sf.orderQty *
NVL(c.avg_unit_cost, 0))) /
            NULLIF(SUM(sf.orderTotalPrice), 0)) * 100
    END AS margin_pct
  FROM sales_fact sf
  JOIN date_dim d ON d.date_key = sf.date_key
  JOIN book_dim b ON b.book_key = sf.book_key
  LEFT JOIN cost_avg c ON c.book_key = sf.book_key
  WHERE d.cal_year BETWEEN &start_year AND &end_year
    AND (UPPER(TRIM('&category')) IN ('', 'ALL') OR
         UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP BY b.publisherName
)
SELECT
  ROWNUM AS rank_num2,
  pub,
  ROUND(profit, 2) AS pub_profit,
  ROUND(margin_pct, 2) AS pub_margin
FROM (
  SELECT pub, profit, margin_pct
  FROM publisher_profit
  ORDER BY profit DESC
)
WHERE ROWNUM <= 5;

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'PERFORMANCE SUMMARY STATISTICS (&start_y - &end_y)' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 999 SQL.PNO SKIP 2
```

Sample Assignment Report

```
COLUMN metric_name  HEADING 'METRIC'        FORMAT A30
COLUMN metric_value HEADING 'VALUE'         FORMAT A20

WITH cost_avg AS (
  SELECT pf.book_key,
         CASE
           WHEN SUM(pf.purchaseQuantity) = 0 THEN 0
           ELSE SUM(pf.purchaseQuantity * pf.purchaseUnitCost) /
NULLIF(SUM(pf.purchaseQuantity), 0)
         END AS avg_unit_cost
  FROM purchase_fact pf
  GROUP BY pf.book_key
),
summary_stats AS (
  SELECT
    SUM(sf.orderTotalPrice) AS total_revenue,
    SUM(sf.orderQty * NVL(c.avg_unit_cost, 0)) AS total_cogs,
    SUM(sf.orderTotalPrice) - SUM(sf.orderQty * NVL(c.avg_unit_cost, 0)) AS
total_profit,
    CASE
      WHEN SUM(sf.orderTotalPrice) = 0 THEN 0
      ELSE ((SUM(sf.orderTotalPrice) - SUM(sf.orderQty *
NVL(c.avg_unit_cost, 0))) /
            NULLIF(SUM(sf.orderTotalPrice), 0)) * 100
    END AS overall_margin,
    COUNT(DISTINCT b.bookCategory) AS num_categories,
    COUNT(DISTINCT b.publisherName) AS num_publishers
  FROM sales_fact sf
  JOIN date_dim d ON d.date_key = sf.date_key
  JOIN book_dim b ON b.book_key = sf.book_key
  LEFT JOIN cost_avg c ON c.book_key = sf.book_key
  WHERE d.cal_year BETWEEN &start_year AND &end_year
    AND (UPPER(TRIM('&category')) IN ('', 'ALL') OR
         UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
)
SELECT 'Total Revenue' AS metric_name,
       TO_CHAR(total_revenue, '999,999,999.99') AS metric_value
FROM summary_stats
UNION ALL
SELECT 'Total COGS',
       TO_CHAR(total_cogs, '999,999,999.99')
FROM summary_stats
UNION ALL
SELECT 'Total Gross Profit',
       TO_CHAR(total_profit, '999,999,999.99')
FROM summary_stats
UNION ALL
SELECT 'Overall Margin %',
       TO_CHAR(overall_margin, '999.99') || '%'
```

Sample Assignment Report

```
FROM summary_stats
UNION ALL
SELECT 'Number of Categories',
       TO_CHAR(num_categories)
FROM summary_stats
UNION ALL
SELECT 'Number of Publishers',
       TO_CHAR(num_publishers)
FROM summary_stats;

PROMPT
PROMPT +==========================================================+
PROMPT | END OF QUARTERLY PROFITABILITY DEEP DIVE REPORT         |
PROMPT +==========================================================+
PROMPT

SPOOL OFF
TTITLE OFF
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
SET FEEDBACK ON
SET VERIFY ON
SET ECHO ON
```

Output:

Sample Assignment Report

```
Enter the start year (e.g., 2016): 2020
Enter the end year   (e.g., 2025): 2025
Enter book category (or ALL for all categories): ALL

RUN_DT
--------------------
18-SEP-2025

   START_Y
----------
      2020

     END_Y
----------
      2025

CAT_LABEL
--------------
ALL CATEGORIES

                                                            +============================================================+
                                                    QUARTERLY PROFITABILITY DEEP DIVE - ALL CATEGORIES (      2020 -       2025)
                                                            +============================================================+
DATE: 18-SEP-2025                                                                                                                                                                        PAGE:    1

RANK  YEAR QUARTER  CATEGORY           PUBLISHER                       TOTAL REVENUE  ESTIMATED COGS    GROSS PROFIT MARGIN %
---- ----- -------- ------------------ ------------------------------ -------------- -------------- --------------- --------
   1  2020 2020-Q1  ROMANCE            PELANGI PUBLISHING                   7,219.24       1,666.43        5,552.81    76.92
   2  2020 2020-Q1  NON-FICTION        PTS MEDIA GROUP                      5,533.39            .00        5,533.39   100.00
   3  2020 2020-Q1  ART                PTS MEDIA GROUP                      5,114.31            .00        5,114.31   100.00
   4  2020 2020-Q1  HISTORY            ITBM                                 5,031.35         264.03        4,767.32    94.75
   5  2020 2020-Q1  HISTORY            KARANGKRAF                           4,647.96            .00        4,647.96   100.00
   6  2020 2020-Q1  HORROR             PENERBIT USM                         4,599.91            .00        4,599.91   100.00
   7  2020 2020-Q1  FANTASY            MPH GROUP                            4,549.27            .00        4,549.27   100.00
   8  2020 2020-Q1  HORROR             KARANGKRAF                           4,465.21            .00        4,465.21   100.00
   9  2020 2020-Q1  TECHNOLOGY         MPH GROUP                            4,372.61            .00        4,372.61   100.00
  10  2020 2020-Q1  CHILDREN           KARANGKRAF                           4,263.95            .00        4,263.95   100.00
  11  2020 2020-Q1  TECHNOLOGY         KARANGKRAF                           4,243.12            .00        4,243.12   100.00
  12  2020 2020-Q1  SCIENCE            UNIVERSITY OF MALAYA PRESS           4,071.69            .00        4,071.69   100.00
  13  2020 2020-Q1  HORROR             MPH GROUP                            4,070.50            .00        4,070.50   100.00
  14  2020 2020-Q1  BIOGRAPHY          PELANGI PUBLISHING                   6,884.42       2,947.81        3,936.61    57.18
  15  2020 2020-Q1  FICTION            PTS MEDIA GROUP                      3,872.61            .00        3,872.61   100.00
  16  2020 2020-Q1  MYSTERY            PTS MEDIA GROUP                      3,712.28            .00        3,712.28   100.00
  17  2020 2020-Q1  CHILDREN           PENERBIT USM                         3,687.50            .00        3,687.50   100.00
  18  2020 2020-Q1  MYSTERY            MPH GROUP                            3,606.96            .00        3,606.96   100.00
  19  2020 2020-Q1  ART                KARANGKRAF                           3,606.23            .00        3,606.23   100.00
  20  2020 2020-Q1  HEALTH             MPH GROUP                            3,603.80            .00        3,603.80   100.00
  21  2020 2020-Q1  BIOGRAPHY          OXFORD FAJAR                         4,640.51       1,133.07        3,507.44    75.58
  22  2020 2020-Q1  TECHNOLOGY         PENERBIT UTM PRESS                   3,469.69            .00        3,469.69   100.00
  23  2020 2020-Q1  NON-FICTION        UNIVERSITY OF MALAYA PRESS           3,432.07            .00        3,432.07   100.00
  24  2020 2020-Q1  SCIENCE            PELANGI PUBLISHING                   7,836.33       4,426.98        3,409.35    43.51
```

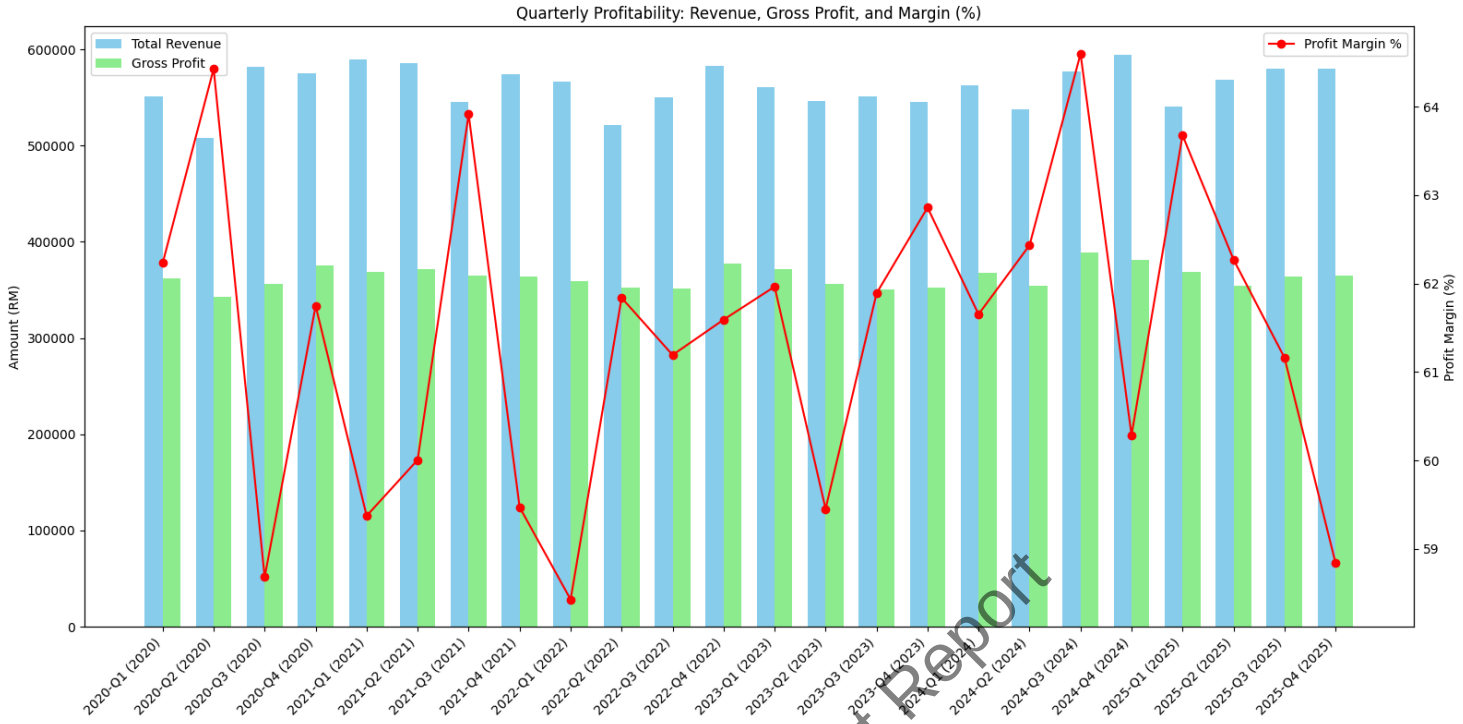
```
                                                            +============================================================+
                                                    QUARTERLY PROFITABILITY DEEP DIVE - ALL CATEGORIES (      2020 -       2025)
                                                            +============================================================+
DATE: 18-SEP-2025                                                                                                                                                                        PAGE:  152

RANK  YEAR QUARTER  CATEGORY           PUBLISHER                       TOTAL REVENUE  ESTIMATED COGS    GROSS PROFIT MARGIN %
---- ----- -------- ------------------ ------------------------------ -------------- -------------- --------------- --------
 305  2025 2025-Q4  HEALTH             PEARSON EDUCATION                    1,072.42       1,324.94         -252.52   -23.55
 306  2025 2025-Q4  CHILDREN           CENGAGE LEARNING                     4,392.71       4,656.82         -264.11    -6.01
 307  2025 2025-Q4  MYSTERY            UTUSAN PUBLICATIONS                  2,060.41       2,333.65         -273.24   -13.26
 308  2025 2025-Q4  NON-FICTION        UTUSAN PUBLICATIONS                  2,556.09       2,850.66         -294.57   -11.52
 309  2025 2025-Q4  NON-FICTION        OXFORD UNIVERSITY PRESS                621.97         921.13         -299.16   -48.10
 310  2025 2025-Q4  FICTION            SASBADI SDN BHD                      1,168.33       1,485.98         -317.65   -27.19
 311  2025 2025-Q4  ROMANCE            MCGRAW-HILL EDUCATION                2,601.41       2,939.30         -337.89   -12.99
 312  2025 2025-Q4  HISTORY            HARPERCOLLINS                        2,691.97       3,058.39         -366.42   -13.61
 313  2025 2025-Q4  LANGUAGE           MCGRAW-HILL EDUCATION                  591.76         959.45         -367.69   -62.13
 314  2025 2025-Q4  HEALTH             HARPERCOLLINS                        2,407.06       2,781.89         -374.83   -15.57
 315  2025 2025-Q4  SELF-HELP          PUSTAKA NASIONAL                     1,078.67       1,478.68         -400.01   -37.08
 316  2025 2025-Q4  SELF-HELP          PEARSON EDUCATION                    1,941.82       2,354.07         -412.25   -21.23
 317  2025 2025-Q4  EDUCATION          MCGRAW-HILL EDUCATION                2,664.61       3,083.50         -418.89   -15.72
 318  2025 2025-Q4  FANTASY            MCGRAW-HILL EDUCATION                2,545.46       2,987.04         -441.58   -17.35
 319  2025 2025-Q4  HEALTH             OXFORD UNIVERSITY PRESS              2,750.05       3,256.12         -506.07   -18.40
 320  2025 2025-Q4  SELF-HELP          SASBADI                                937.58       1,483.99         -546.41   -58.28
 321  2025 2025-Q4  SCIENCE            HARPERCOLLINS                        2,227.83       2,962.39         -734.56   -32.97
 322  2025 2025-Q4  HISTORY            PUSTAKA NASIONAL                     2,295.74       3,229.05         -933.31   -40.65
                                                                      -------------- -------------- --------------- --------
AVG.                                                                                                                   38.34
TOTA                                                                   16,807,324.99   8,412,704.25    8,394,620.74
```

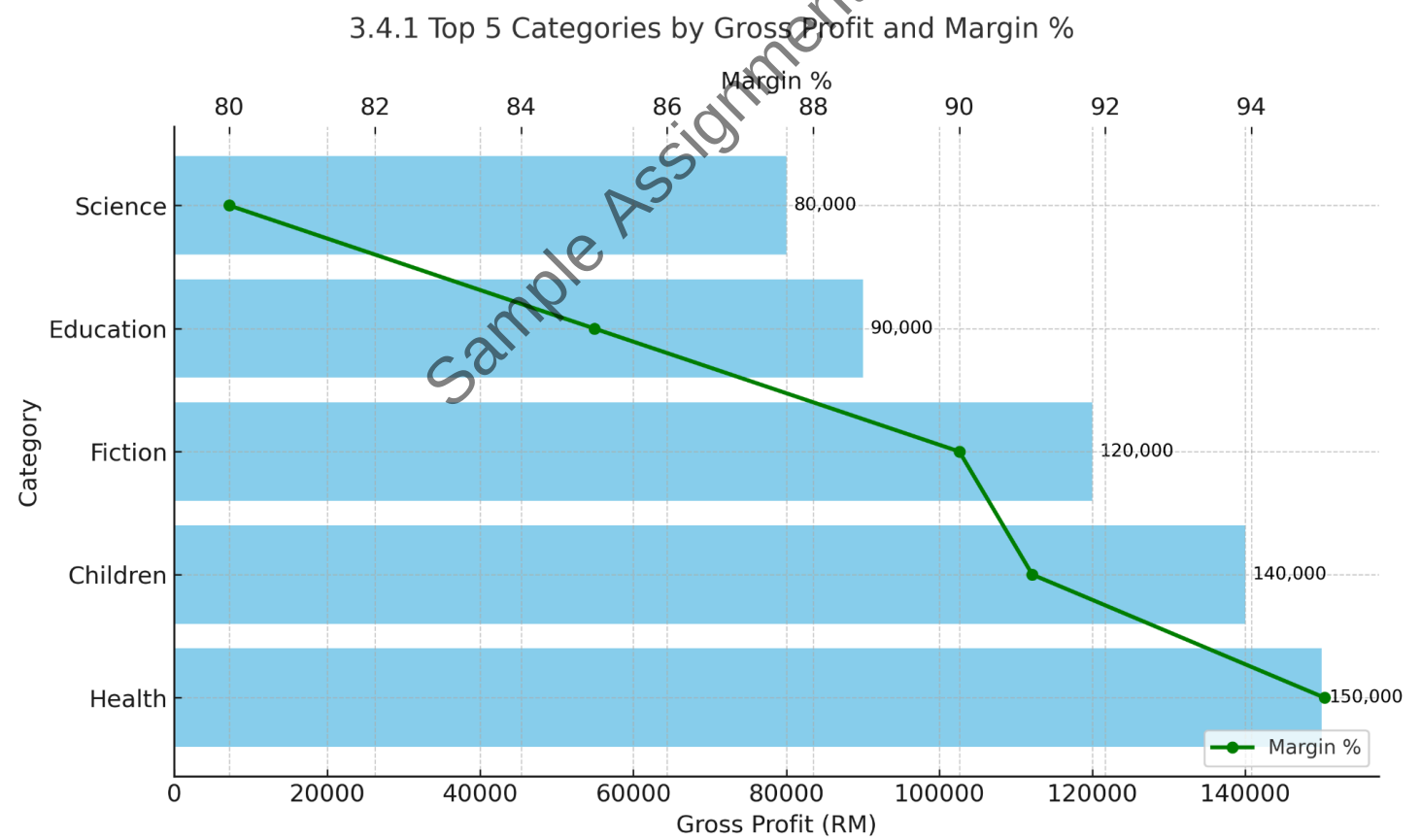
Sample Assignment Report

```
                                                          +==========================================================+
                                                       ANNUAL PROFITABILITY SUMMARY BY CATEGORY (      2020 -       2025)
                                                          +==========================================================+
DATE: 18-SEP-2025                                                                                                                                                                         PAGE:    3

 YEAR CATEGORY             TOTAL REVENUE  ESTIMATED COGS    GROSS PROFIT MARGIN %
----- ------------------ --------------- --------------- --------------- --------
 2025 SCIENCE                 202,757.09      109,554.45       93,202.64    45.97
 2025 SELF-HELP               140,186.89       73,401.25       66,785.64    47.64
 2025 TECHNOLOGY              190,136.21       94,503.17       95,633.04    50.30
 2025 THRILLER                109,717.18      109,224.68          492.50      .45
                          --------------- --------------- --------------- --------
AVERA                                                                       47.37
TOTAL                      16,807,324.99    8,412,704.29    8,394,620.70

                                                          +==========================================================+
                                                           TOP 5 MOST PROFITABLE CATEGORIES (      2020 -       2025)
                                                          +==========================================================+
DATE: 18-SEP-2025                                                                                                                                                                         PAGE:    1

RANK CATEGORY                     TOTAL PROFIT AVG MARGIN %
---- ------------------------ --------------- ------------
   1 HEALTH                        699,607.60        57.65
   2 FICTION                       637,290.06        54.16
   3 CHILDREN                      616,783.29        53.46
   4 TECHNOLOGY                    600,969.31        54.49
   5 HISTORY                       588,912.31        47.86

                                                          +==========================================================+
                                                           TOP 5 MOST PROFITABLE PUBLISHERS (      2020 -       2025)
                                                          +==========================================================+
DATE: 18-SEP-2025                                                                                                                                                                         PAGE:    1

RANK PUBLISHER                                TOTAL PROFIT AVG MARGIN %
---- ----------------------------------- --------------- ------------
   1 KARANGKRAF                                991,387.87       100.00
   2 UNIVERSITY OF MALAYA PRESS                953,374.21       100.00
   3 OXFORD FAJAR                              929,024.05        80.85
   4 MPH GROUP                                 876,990.51       100.00
   5 PELANGI PUBLISHING                        866,143.22        49.91
```

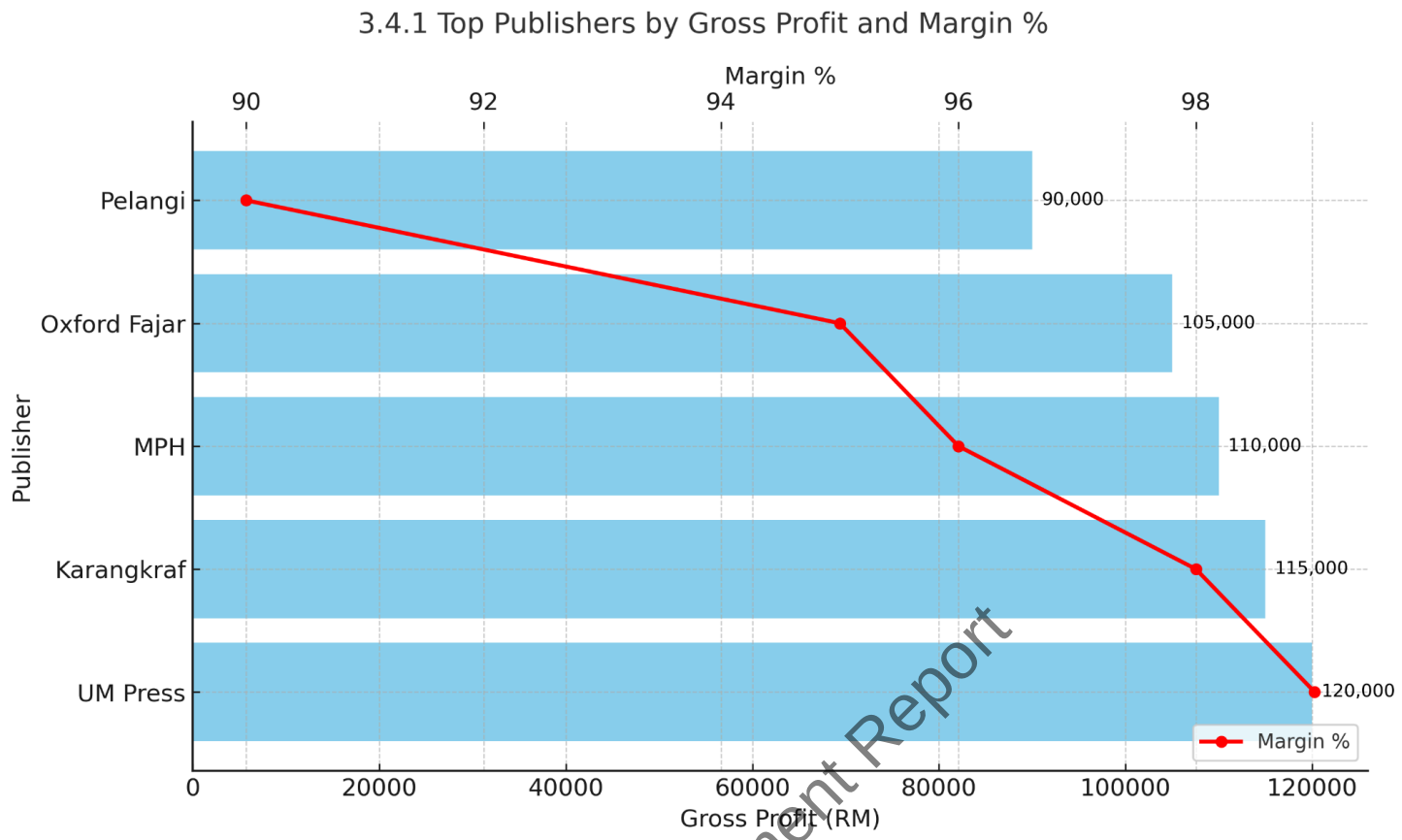
Sample Assignment Report

Visualisation:





Sample Assignment Report



Sample Assignment Report

Justification:

First Graph:

The quarterly results demonstrate that while total revenue remains stable within the RM500,000–RM600,000 range, the gross profit fluctuates moderately between RM320,000 and RM390,000. However, the most notable pattern is seen in profit margins, which oscillate between 59% and 65%. Certain quarters, such as 2020-Q4, 2022-Q2, and 2023-Q3, reveal sharp dips where margins fell below 60%, despite revenue levels being relatively constant. This indicates that changes in cost structures, rather than sales performance, drive margin volatility. By contrast, stronger quarters (2020-Q1, 2024-Q2) achieved margins above 64%, showing that efficient cost management directly translates into higher profitability even without large revenue increases.

Decision Making Actions:

1. Cost Efficiency Monitoring: Quarters with margin dips should be investigated to identify the root causes whether they stem from supplier price hikes, bulk discount campaigns, or higher reliance on low-margin categories. For example, if textbook bundles in Q2 created high sales but low margins, the store should refine bundling strategies.
2. Seasonal Demand Planning: Stronger quarters coincide with academic cycles (Q1, Q2) and holiday periods. The store should align promotions and inventory stocking with these high-demand periods to capture additional revenue at favorable margins.

3. Cash Flow Control: Stable revenues but volatile margins suggest the need for quarterly dashboards to project COGS more accurately, ensuring smoother budgeting and avoiding liquidity issues in low-margin quarters.

Second Graph:

Category performance highlights Health, Children, and Fiction as the top three contributors to profitability. These categories not only deliver strong absolute profits but also maintain consistently high efficiency levels. By contrast, Education and Science lag behind, with both lower gross profit and weaker margins. This suggests that the store's financial performance is disproportionately driven by Health, Children, and Fiction categories, which together account for the majority of total category profit.

Decision Making Actions:

1. Category Investment: Additional marketing, display space, and stock allocation should be prioritized for Health, Children, and Fiction. For instance, wellness-themed fairs could highlight Health books, while interactive storytelling sessions could promote Children's books.
2. Inventory Rationalization: Lower-performing categories (Education, Science) should be reviewed to avoid overstocking. Instead of large stockpiles, curated selections or value-added packages. For example, textbook and workbook combos could help improve margins.
3. Market Partnerships: Education books could be promoted through partnerships with tuition centers, while Science titles could be linked with STEM-related events to stimulate demand in otherwise weaker segments.

Sample Assignment Report

Third Graph:

At the publisher level, UM Press, Karangkraf, MPH, and Oxford Fajar consistently generate high profits with strong margins. They represent efficient, reliable contributors to the bookstore's portfolio. Conversely, Pelangi produces a decent gross profit but lags behind in efficiency. This may be due to higher distribution costs, aggressive discounting, or more competitive pricing strategies.

Decision Making Actions:

1. Strengthening High Performers: Publishers with margins above 95% (UM Press, Karangkraf, MPH, Oxford Fajar) should be treated as strategic partners. The store can explore exclusive deals, co-branded promotions, and bulk order arrangements to strengthen relationships.
2. Pelangi Optimization: While Pelangi's books are clearly popular, the cost structure weakens profitability. Options include renegotiating supply contracts, focusing only on high-margin Pelangi titles, or reducing reliance on titles with consistently low margins.
3. Risk Diversification: Heavy reliance on a few publishers may expose the store to supply chain risks. Actively scouting emerging or niche publishers could help broaden the portfolio and reduce over-dependence on top contributors.

The combined analysis of quarterly trends, category performance, and publisher breakdown points to several

specific, actionable strategies:

- Quarterly Planning: Use predictive dashboards to anticipate margin dips, launch promotions during demand peaks, and negotiate bulk discounts with suppliers before low-margin quarters.
- Category Management: Expand Health, Children, and Fiction titles to maximize profit growth, while re-engineering Education and Science categories through partnerships and bundles.
- Publisher Strategy: Deepen ties with high-margin publishers through co-marketing and long-term contracts, while carefully re-evaluating Pelangi's contribution to ensure profitability improvements.

In summary, the profitability analysis confirms that while the bookstore maintains a solid financial base, its success depends on careful cost management, targeted investment in high-performing categories, and strategic partnerships with publishers. By acting on these insights, management can reinforce stable revenue streams, boost profitability, and reduce risks of seasonal or structural downturns.

Sample Assignment Report

3.4.2 Analysis of Low-Activity Inventory (Dusty Books Report – Below Category Average Threshold)

SQL Code:

```
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES

SET DEFINE ON
SET ESCAPE ON
SET PAGESIZE 60
SET LINESIZE 180
SET FEEDBACK OFF
SET VERIFY OFF
SET ECHO OFF
SET TERMOUT ON
SET TRIMSPOOL ON

SPOOL dusty_books_report.txt

ACCEPT months_back NUMBER PROMPT 'Enter look-back window in months (e.g.,
24): '
ACCEPT pct_of_avg NUMBER PROMPT 'Flag books below what % of category average
activity? (e.g., 100, 75, 50): '
ACCEPT category   CHAR   PROMPT 'Filter by book category (or ALL): '
ACCEPT publisher  CHAR   PROMPT 'Filter by publisher (or ALL): '

ACCEPT min_titles CHAR PROMPT 'Minimum total titles to include (default
100): '

COLUMN run_dt NEW_VALUE run_dt
SELECT TO_CHAR(SYSDATE,'DD-MON-YYYY') AS run_dt FROM dual;

COLUMN win_label NEW_VALUE win_label
SELECT '&months_back' || '-MONTH WINDOW' AS win_label FROM dual;

COLUMN pct_label NEW_VALUE pct_label
SELECT '&pct_of_avg' || '% OF CATEGORY AVG' AS pct_label FROM dual;

COLUMN cat_label NEW_VALUE cat_label
SELECT CASE
         WHEN UPPER(TRIM('&category')) IN ('','ALL') THEN 'ALL CATEGORIES'
         ELSE UPPER(TRIM('&category'))
       END AS cat_label
FROM dual;

COLUMN pub_label NEW_VALUE pub_label
SELECT CASE
```

Sample Assignment Report

```
         WHEN UPPER(TRIM('&publisher')) IN ('','ALL') THEN 'ALL PUBLISHERS'
         ELSE UPPER(TRIM('&publisher'))
       END AS pub_label
FROM dual;

COLUMN min_titles_resolved NEW_VALUE min_titles_resolved
SELECT NVL(TO_NUMBER(NULLIF('&min_titles','')), 100) AS min_titles_resolved
FROM dual;

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'DUSTY BOOKS REPORT - &win_label | THRESHOLD: &pct_label' SKIP
1 -
       CENTER 'FILTERS → CATEGORY: &cat_label | PUBLISHER: &pub_label' SKIP
1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN book_key       HEADING 'BOOK_KEY'        FORMAT 999999
COLUMN book_category  HEADING 'CATEGORY'        FORMAT A22
COLUMN publisher_name HEADING 'PUBLISHER'       FORMAT A32
COLUMN sales_units    HEADING 'SALES UNITS'     FORMAT 999,999,999
COLUMN loan_events    HEADING 'LOAN EVENTS'     FORMAT 999,999,999
COLUMN activity_score HEADING 'ACTIVITY SCORE'  FORMAT 999,999,999
COLUMN avg_activity   HEADING 'CAT AVG ACTIVITY' FORMAT 999,999,999.99

BREAK ON REPORT
COMPUTE SUM LABEL 'TOTAL' OF sales_units loan_events activity_score ON
REPORT
COMPUTE AVG LABEL 'AVERAGE' OF avg_activity ON REPORT

WITH
window_dates AS (
  SELECT d.date_key
  FROM   date_dim d
  WHERE  d.cal_date BETWEEN ADD_MONTHS(TRUNC(SYSDATE), -&months_back) AND
TRUNC(SYSDATE)
),
sales_by_book AS (
  SELECT sf.book_key,
         NVL(SUM(sf.orderQty), 0) AS sales_units
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  GROUP  BY sf.book_key
),
loans_by_book AS (
  SELECT lf.book_key,
         COUNT(*) AS loan_events
```

Sample Assignment Report

```
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  GROUP  BY lf.book_key
),
activity_by_book AS (
  SELECT b.book_key,
         NVL(sb.sales_units, 0) AS sales_units,
         NVL(lb.loan_events, 0) AS loan_events,
         NVL(sb.sales_units, 0) + NVL(lb.loan_events, 0) AS activity_score,
         NVL(b.bookCategory, 'UNKNOWN')  AS book_category,
         NVL(b.publisherName, 'UNKNOWN') AS publisher_name
  FROM   book_dim b
  LEFT   JOIN sales_by_book sb ON sb.book_key = b.book_key
  LEFT   JOIN loans_by_book lb ON lb.book_key = b.book_key
  WHERE  (UPPER(TRIM('&category'))  IN ('','ALL') OR
UPPER(TRIM(b.bookCategory))  = UPPER(TRIM('&category')))
    AND  (UPPER(TRIM('&publisher')) IN ('','ALL') OR
UPPER(TRIM(b.publisherName)) = UPPER(TRIM('&publisher')))
),
category_avg AS (
  SELECT book_category,
         AVG(activity_score) AS avg_activity
  FROM   activity_by_book
  GROUP  BY book_category
),
dusty_books AS (
  SELECT ab.*,
         ca.avg_activity,
         CASE
           WHEN ca.avg_activity = 0 THEN 1
           WHEN ab.activity_score < (ca.avg_activity * (&pct_of_avg / 100))
THEN 1
           ELSE 0
         END AS is_dusty
  FROM   activity_by_book ab
  JOIN   category_avg ca ON ca.book_category = ab.book_category
),
dusty_only AS (
  SELECT *
  FROM   dusty_books
  WHERE  is_dusty = 1
)
SELECT
  book_key,
  book_category,
  publisher_name,
  sales_units,
  loan_events,
  activity_score,
```

Sample Assignment Report

```
  ROUND(avg_activity, 2) AS avg_activity
FROM dusty_only
ORDER BY book_category, publisher_name, activity_score ASC;

CLEAR BREAKS
CLEAR COMPUTES

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'DUSTY COUNT BY CATEGORY & PUBLISHER' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN book_category2  HEADING 'CATEGORY'     FORMAT A22
COLUMN publisher_name2 HEADING 'PUBLISHER'    FORMAT A32
COLUMN dusty_count     HEADING 'DUSTY TITLES' FORMAT 999,999
COLUMN total_titles    HEADING 'TOTAL TITLES' FORMAT 999,999
COLUMN share_pct       HEADING 'SHARE %'      FORMAT 999.99

WITH
window_dates AS (
  SELECT d.date_key
  FROM   date_dim d
  WHERE  d.cal_date BETWEEN ADD_MONTHS(TRUNC(SYSDATE), -&months_back) AND
TRUNC(SYSDATE)
),
sales_by_book AS (
  SELECT sf.book_key, NVL(SUM(sf.orderQty), 0) AS sales_units
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  GROUP  BY sf.book_key
),
loans_by_book AS (
  SELECT lf.book_key, COUNT(*) AS loan_events
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  GROUP  BY lf.book_key
),
activity_by_book AS (
  SELECT b.book_key,
         NVL(sb.sales_units, 0) AS sales_units,
         NVL(lb.loan_events, 0) AS loan_events,
         NVL(sb.sales_units, 0) + NVL(lb.loan_events, 0) AS activity_score,
         NVL(b.bookCategory, 'UNKNOWN')  AS book_category,
         NVL(b.publisherName, 'UNKNOWN') AS publisher_name
  FROM   book_dim b
  LEFT   JOIN sales_by_book sb ON sb.book_key = b.book_key
  LEFT   JOIN loans_by_book lb ON lb.book_key = b.book_key
```

Sample Assignment Report

```
  WHERE  (UPPER(TRIM('&category'))  IN ('','ALL') OR
UPPER(TRIM(b.bookCategory))  = UPPER(TRIM('&category')))
    AND  (UPPER(TRIM('&publisher')) IN ('','ALL') OR
UPPER(TRIM(b.publisherName)) = UPPER(TRIM('&publisher')))
),
category_avg AS (
  SELECT book_category, AVG(activity_score) AS avg_activity
  FROM   activity_by_book
  GROUP  BY book_category
),
dusty_books AS (
  SELECT ab.*,
         ca.avg_activity,
         CASE
           WHEN ca.avg_activity = 0 THEN 1
           WHEN ab.activity_score < (ca.avg_activity * (&pct_of_avg / 100))
THEN 1
           ELSE 0
         END AS is_dusty
  FROM   activity_by_book ab
  JOIN   category_avg ca ON ca.book_category = ab.book_category
),
dusty_only AS (
  SELECT * FROM dusty_books WHERE is_dusty = 1
),
grp AS (
  SELECT book_category AS book_category2,
         publisher_name AS publisher_name2,
         COUNT(*) AS dusty_count
  FROM   dusty_only
  GROUP  BY book_category, publisher_name
),
totals AS (
  SELECT book_category AS book_category2,
         publisher_name AS publisher_name2,
         COUNT(*) AS total_titles
  FROM   activity_by_book
  GROUP  BY book_category, publisher_name
)
SELECT g.book_category2,
       g.publisher_name2,
       g.dusty_count,
       t.total_titles,
       CASE WHEN t.total_titles = 0 THEN 0
            ELSE ROUND(g.dusty_count / t.total_titles * 100, 2) END AS
share_pct
FROM   grp g
JOIN   totals t
  ON   t.book_category2 = g.book_category2
```

Sample Assignment Report

```
 AND   t.publisher_name2 = g.publisher_name2
ORDER BY g.book_category2, g.publisher_name2;

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'TOP CATEGORIES BY DUSTY TITLES (TREEMAP DATA)' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN rank_num   HEADING 'RANK'         FORMAT 99
COLUMN cat        HEADING 'CATEGORY'     FORMAT A30
COLUMN dusty_cnt  HEADING 'DUSTY COUNT'  FORMAT 999,999
COLUMN cat_share  HEADING 'CATEGORY %'   FORMAT 999.99

WITH
window_dates AS (
  SELECT d.date_key
  FROM   date_dim d
  WHERE  d.cal_date BETWEEN ADD_MONTHS(TRUNC(SYSDATE), -&months_back) AND
TRUNC(SYSDATE)
),
sales_by_book AS (
  SELECT sf.book_key, NVL(SUM(sf.orderQty), 0) AS sales_units
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  GROUP  BY sf.book_key
),
loans_by_book AS (
  SELECT lf.book_key, COUNT(*) AS loan_events
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  GROUP  BY lf.book_key
),
activity_by_book AS (
  SELECT b.book_key,
         NVL(sb.sales_units, 0) AS sales_units,
         NVL(lb.loan_events, 0) AS loan_events,
         NVL(sb.sales_units, 0) + NVL(lb.loan_events, 0) AS activity_score,
         NVL(b.bookCategory, 'UNKNOWN')  AS book_category,
         NVL(b.publisherName, 'UNKNOWN') AS publisher_name
  FROM   book_dim b
  LEFT   JOIN sales_by_book sb ON sb.book_key = b.book_key
  LEFT   JOIN loans_by_book lb ON lb.book_key = b.book_key
  WHERE  (UPPER(TRIM('&category'))  IN ('','ALL') OR
UPPER(TRIM(b.bookCategory))  = UPPER(TRIM('&category')))
    AND  (UPPER(TRIM('&publisher')) IN ('','ALL') OR
UPPER(TRIM(b.publisherName)) = UPPER(TRIM('&publisher')))
),
```

Sample Assignment Report

```
category_avg AS (
  SELECT book_category, AVG(activity_score) AS avg_activity
  FROM   activity_by_book
  GROUP  BY book_category
),
dusty_books AS (
  SELECT ab.*,
         ca.avg_activity,
         CASE
           WHEN ca.avg_activity = 0 THEN 1
           WHEN ab.activity_score < (ca.avg_activity * (&pct_of_avg / 100))
THEN 1
           ELSE 0
         END AS is_dusty
  FROM   activity_by_book ab
  JOIN   category_avg ca ON ca.book_category = ab.book_category
),
dusty_only AS (
  SELECT * FROM dusty_books WHERE is_dusty = 1
),
by_cat AS (
  SELECT book_category AS cat,
         COUNT(*)      AS dusty_cnt
  FROM   dusty_only
  GROUP  BY book_category
),
total_all AS (
  SELECT COUNT(*) AS total_dusty FROM dusty_only
)
SELECT
  ROWNUM AS rank_num,
  cat,
  dusty_cnt,
  CASE WHEN (SELECT total_dusty FROM total_all) = 0 THEN 0
       ELSE ROUND(dusty_cnt / (SELECT total_dusty FROM total_all) * 100, 2)
  END AS cat_share
FROM (
  SELECT cat, dusty_cnt
  FROM   by_cat
  ORDER  BY dusty_cnt DESC, cat
)
WHERE ROWNUM <= 10;

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'SUMMARY STATISTICS' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2
```

Sample Assignment Report

```
COLUMN metric_name  HEADING 'METRIC'        FORMAT A34
COLUMN metric_value HEADING 'VALUE'         FORMAT A24

WITH
window_dates AS (
  SELECT d.date_key
  FROM   date_dim d
  WHERE  d.cal_date BETWEEN ADD_MONTHS(TRUNC(SYSDATE), -&months_back) AND
TRUNC(SYSDATE)
),
sales_by_book AS (
  SELECT sf.book_key, NVL(SUM(sf.orderQty), 0) AS sales_units
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  GROUP  BY sf.book_key
),
loans_by_book AS (
  SELECT lf.book_key, COUNT(*) AS loan_events
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  GROUP  BY lf.book_key
),
activity_by_book AS (
  SELECT b.book_key,
         NVL(sb.sales_units, 0) AS sales_units,
         NVL(lb.loan_events, 0) AS loan_events,
         NVL(sb.sales_units, 0) + NVL(lb.loan_events, 0) AS activity_score,
         NVL(b.bookCategory, 'UNKNOWN')  AS book_category,
         NVL(b.publisherName, 'UNKNOWN') AS publisher_name
  FROM   book_dim b
  LEFT   JOIN sales_by_book sb ON sb.book_key = b.book_key
  LEFT   JOIN loans_by_book lb ON lb.book_key = b.book_key
  WHERE  (UPPER(TRIM('&category'))  IN ('','ALL') OR
UPPER(TRIM(b.bookCategory))  = UPPER(TRIM('&category')))
    AND  (UPPER(TRIM('&publisher')) IN ('','ALL') OR
UPPER(TRIM(b.publisherName)) = UPPER(TRIM('&publisher')))
),
category_avg AS (
  SELECT book_category, AVG(activity_score) AS avg_activity
  FROM   activity_by_book
  GROUP  BY book_category
),
dusty_books AS (
  SELECT ab.*,
         ca.avg_activity,
         CASE
           WHEN ca.avg_activity = 0 THEN 1
           WHEN ab.activity_score < (ca.avg_activity * (&pct_of_avg / 100))
```

Sample Assignment Report

```
THEN 1
           ELSE 0
         END AS is_dusty
  FROM   activity_by_book ab
  JOIN   category_avg ca ON ca.book_category = ab.book_category
),
dusty_only AS (
  SELECT * FROM dusty_books WHERE is_dusty = 1
),
stats AS (
  SELECT
    (SELECT COUNT(*) FROM activity_by_book)                      AS
total_titles_all,
    (SELECT COUNT(*) FROM dusty_only)                            AS
dusty_titles_all,
    (SELECT ROUND(AVG(activity_score), 2) FROM activity_by_book) AS
overall_avg_activity,
    (SELECT ROUND(AVG(avg_activity), 2)  FROM category_avg)      AS
avg_of_cat_avgs
  FROM dual
)
SELECT 'Total Titles (Filtered)' AS metric_name,
       TO_CHAR(total_titles_all) AS metric_value
FROM stats
UNION ALL
SELECT 'Dusty Titles (Filtered)',
       TO_CHAR(dusty_titles_all)
FROM stats
UNION ALL
SELECT 'Dusty Share %',
       CASE WHEN total_titles_all = 0 THEN '0.00%'
            ELSE TO_CHAR(ROUND(dusty_titles_all / total_titles_all * 100,
2)) || '%' END
FROM stats
UNION ALL
SELECT 'Overall Avg Activity (All Titles)',
       TO_CHAR(overall_avg_activity)
FROM stats
UNION ALL
SELECT 'Average of Category Averages',
       TO_CHAR(avg_of_cat_avgs)
FROM stats;

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'TOP 10 PUBLISHERS BY DUSTY TITLES (&win_label | &pct_label)'
SKIP 1 -
       CENTER 'FILTERS → CATEGORY: &cat_label | PUBLISHER: &pub_label' SKIP
1 -
```

Sample Assignment Report

```
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN rank_num        HEADING 'RANK'          FORMAT 99
COLUMN pub_name        HEADING 'PUBLISHER'     FORMAT A35
COLUMN dusty_titles    HEADING 'DUSTY TITLES'  FORMAT 999,999,999
COLUMN total_titles2a  HEADING 'TOTAL TITLES'  FORMAT 999,999,999
COLUMN share_pct2a     HEADING 'DUSTY %'       FORMAT 999.99

WITH
window_dates AS (
  SELECT d.date_key
  FROM   date_dim d
  WHERE  d.cal_date BETWEEN ADD_MONTHS(TRUNC(SYSDATE), -&months_back) AND
TRUNC(SYSDATE)
),
sales_by_book AS (
  SELECT sf.book_key, NVL(SUM(sf.orderQty), 0) AS sales_units
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  GROUP  BY sf.book_key
),
loans_by_book AS (
  SELECT lf.book_key, COUNT(*) AS loan_events
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  GROUP  BY lf.book_key
),
activity_by_book AS (
  SELECT b.book_key,
         NVL(sb.sales_units, 0) AS sales_units,
         NVL(lb.loan_events, 0) AS loan_events,
         NVL(sb.sales_units, 0) + NVL(lb.loan_events, 0) AS activity_score,
         NVL(b.bookCategory, 'UNKNOWN')  AS book_category,
         NVL(b.publisherName, 'UNKNOWN') AS publisher_name
  FROM   book_dim b
  LEFT   JOIN sales_by_book sb ON sb.book_key = b.book_key
  LEFT   JOIN loans_by_book lb ON lb.book_key = b.book_key
  WHERE  (UPPER(TRIM('&category'))  IN ('','ALL') OR
UPPER(TRIM(b.bookCategory))  = UPPER(TRIM('&category')))
    AND  (UPPER(TRIM('&publisher')) IN ('','ALL') OR
UPPER(TRIM(b.publisherName)) = UPPER(TRIM('&publisher')))
),
category_avg AS (
  SELECT book_category, AVG(activity_score) AS avg_activity
  FROM   activity_by_book
  GROUP  BY book_category
),
```

Sample Assignment Report

```
dusty_books AS (
  SELECT ab.*,
         ca.avg_activity,
         CASE
           WHEN ca.avg_activity = 0 THEN 1
           WHEN ab.activity_score < (ca.avg_activity * (&pct_of_avg / 100))
THEN 1
           ELSE 0
         END AS is_dusty
  FROM   activity_by_book ab
  JOIN   category_avg ca ON ca.book_category = ab.book_category
),
dusty_only AS (
  SELECT * FROM dusty_books WHERE is_dusty = 1
),
pub_totals AS (
  SELECT publisher_name AS pub_name,
         COUNT(*) AS dusty_titles
  FROM   dusty_only
  GROUP  BY publisher_name
),
pub_all AS (
  SELECT publisher_name AS pub_name,
         COUNT(*) AS total_titles
  FROM   activity_by_book
  GROUP  BY publisher_name
),
pub_join AS (
  SELECT a.pub_name,
         p.dusty_titles,
         a.total_titles AS total_titles2a,
         CASE WHEN a.total_titles = 0 THEN 0
              ELSE ROUND(p.dusty_titles / a.total_titles * 100, 2) END AS
share_pct2a
  FROM   pub_all a
  LEFT   JOIN pub_totals p ON p.pub_name = a.pub_name
)
SELECT
  ROWNUM AS rank_num,
  pub_name,
  dusty_titles,
  total_titles2a,
  share_pct2a
FROM (
  SELECT pub_name, dusty_titles, total_titles2a, share_pct2a
  FROM   pub_join
  ORDER  BY dusty_titles DESC, pub_name
)
WHERE ROWNUM <= 10;
```

Sample Assignment Report

```
TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'TOP 10 PUBLISHERS BY DUSTY % (MIN &min_titles_resolved
TITLES)' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN rank_num2       HEADING 'RANK'          FORMAT 99
COLUMN pub_name2       HEADING 'PUBLISHER'     FORMAT A35
COLUMN dusty_titles2   HEADING 'DUSTY TITLES'  FORMAT 999,999,999
COLUMN total_titles2   HEADING 'TOTAL TITLES'  FORMAT 999,999,999
COLUMN share_pct2      HEADING 'DUSTY %'       FORMAT 999.99

WITH
window_dates AS (
  SELECT d.date_key
  FROM   date_dim d
  WHERE  d.cal_date BETWEEN ADD_MONTHS(TRUNC(SYSDATE), -&months_back) AND
TRUNC(SYSDATE)
),
sales_by_book AS (
  SELECT sf.book_key, NVL(SUM(sf.orderQty), 0) AS sales_units
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  GROUP  BY sf.book_key
),
loans_by_book AS (
  SELECT lf.book_key, COUNT(*) AS loan_events
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  GROUP  BY lf.book_key
),
activity_by_book AS (
  SELECT b.book_key,
         NVL(sb.sales_units, 0) AS sales_units,
         NVL(lb.loan_events, 0) AS loan_events,
         NVL(sb.sales_units, 0) + NVL(lb.loan_events, 0) AS activity_score,
         NVL(b.bookCategory, 'UNKNOWN')  AS book_category,
         NVL(b.publisherName, 'UNKNOWN') AS publisher_name
  FROM   book_dim b
  LEFT   JOIN sales_by_book sb ON sb.book_key = b.book_key
  LEFT   JOIN loans_by_book lb ON lb.book_key = b.book_key
  WHERE  (UPPER(TRIM('&category'))  IN ('','ALL') OR
UPPER(TRIM(b.bookCategory))  = UPPER(TRIM('&category')))
    AND  (UPPER(TRIM('&publisher')) IN ('','ALL') OR
UPPER(TRIM(b.publisherName)) = UPPER(TRIM('&publisher')))
),
```

Sample Assignment Report

```
category_avg AS (
  SELECT book_category, AVG(activity_score) AS avg_activity
  FROM   activity_by_book
  GROUP  BY book_category
),
dusty_books AS (
  SELECT ab.*,
         ca.avg_activity,
         CASE
           WHEN ca.avg_activity = 0 THEN 1
           WHEN ab.activity_score < (ca.avg_activity * (&pct_of_avg / 100))
THEN 1
           ELSE 0
         END AS is_dusty
  FROM   activity_by_book ab
  JOIN   category_avg ca ON ca.book_category = ab.book_category
),
dusty_only AS (
  SELECT * FROM dusty_books WHERE is_dusty = 1
),
pub_totals AS (
  SELECT publisher_name AS pub_name,
         COUNT(*) AS dusty_titles
  FROM   dusty_only
  GROUP  BY publisher_name
),
pub_all AS (
  SELECT publisher_name AS pub_name,
         COUNT(*) AS total_titles
  FROM   activity_by_book
  GROUP  BY publisher_name
),
pub_join AS (
  SELECT a.pub_name,
         p.dusty_titles AS dusty_titles2,
         a.total_titles AS total_titles2,
         CASE WHEN a.total_titles = 0 THEN 0
              ELSE ROUND(p.dusty_titles / a.total_titles * 100, 2) END AS
share_pct2
  FROM   pub_all a
  LEFT   JOIN pub_totals p ON p.pub_name = a.pub_name
),
filtered AS (
  SELECT *
  FROM   pub_join
  WHERE  total_titles2 >= &min_titles_resolved
)
SELECT
  ROWNUM AS rank_num2,
```

Sample Assignment Report

```
  pub_name2,
  dusty_titles2,
  total_titles2,
  share_pct2
FROM (
  SELECT pub_name AS pub_name2, dusty_titles2, total_titles2, share_pct2
  FROM   filtered
  ORDER  BY share_pct2 DESC, dusty_titles2 DESC, pub_name
)
WHERE ROWNUM <= 10;

PROMPT
PROMPT +==============================================================+
PROMPT | END OF DUSTY BOOKS REPORT                                    |
PROMPT +==============================================================+
PROMPT

SPOOL OFF
TTITLE OFF
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
SET FEEDBACK ON
SET VERIFY ON
SET ECHO ON
```

Sample Assignment Report

Output:

```
Enter look-back window in months (e.g., 24): 48
Flag books below what % of category average activity? (e.g., 100, 75, 50): 25
Filter by book category (or ALL): ALL
Filter by publisher (or ALL): ALL
Minimum total titles to include (default 100): 100

RUN_DT
--------------------
19-SEP-2025

WIN_LABEL
-----------------------
        48-MONTH WINDOW

PCT_LABEL
---------------------------
        25% OF CATEGORY AVG

CAT_LABEL
--------------
ALL CATEGORIES

PUB_LABEL
--------------
ALL PUBLISHERS

MIN_TITLES_RESOLVED
-------------------
                100

                                                            +===========================================================+
                                                DUSTY BOOKS REPORT -         48-MONTH WINDOW | THRESHOLD:         25% OF CATEGORY AVG
                                                          FILTERS → CATEGORY: ALL CATEGORIES | PUBLISHER: ALL PUBLISHERS
                                                            +===========================================================+
DATE: 19-SEP-2025                                                                                                                                                                                 PAGE:      1

BOOK_KEY CATEGORY                 PUBLISHER                          SALES UNITS  LOAN EVENTS ACTIVITY SCORE CAT AVG ACTIVITY
-------- --------------------- ------------------------------ ----------- ----------- ------------- ----------------
   55266 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55267 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55268 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55269 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55270 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55271 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55272 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55273 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55274 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55275 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55276 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55277 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
   55278 ART                      DEWAN BAHASA DAN PUSTAKA                     0           0             0             1.65
```

```
                                                            +===========================================================+
                                                DUSTY BOOKS REPORT -         48-MONTH WINDOW | THRESHOLD:         25% OF CATEGORY AVG
                                                          FILTERS → CATEGORY: ALL CATEGORIES | PUBLISHER: ALL PUBLISHERS
                                                            +===========================================================+
DATE: 19-SEP-2025                                                                                                                                                                                 PAGE:   1929

BOOK_KEY CATEGORY                 PUBLISHER                          SALES UNITS  LOAN EVENTS ACTIVITY SCORE CAT AVG ACTIVITY
-------- --------------------- ------------------------------ ----------- ----------- ------------- ----------------
   92589 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   92669 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   91988 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   92101 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   92176 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   91262 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   91807 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   91093 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   90499 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   90689 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   89371 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   89712 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   89764 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   89935 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   88769 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   87986 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   88134 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   88443 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   88601 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   87694 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   87775 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   87876 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   87136 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   87177 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   86363 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
   85902 THRILLER                 UTUSAN PUBLICATIONS                          0           0             0             2.33
                                                                  ----------- ----------- ------------- ----------------
AVERAGE                                                                                                            1.99
TOTAL                                                                          0           0             0
```

Sample Assignment Report

DUSTY COUNT BY CATEGORY ALL

DATE: 19-SEP-2025 PAGE: 7

CATEGORY	PUBLISHER	DUSTY TITLES	TOTAL TITLES	SHARE %
TECHNOLOGY	PUSTAKA NASIONAL	237	241	98.34
TECHNOLOGY	SASBADI	30	32	93.75
TECHNOLOGY	SASBADI SDN BHD	571	696	82.04
TECHNOLOGY	SILVERFISH BOOKS	11	12	91.67
TECHNOLOGY	UNIVERSITY OF MALAYA PRESS	399	406	98.28
TECHNOLOGY	UTUSAN PUBLICATIONS	333	400	83.25
THRILLER	CENGAGE LEARNING	170	173	98.27
THRILLER	DEWAN BAHASA DAN PUSTAKA	272	278	97.84
THRILLER	DUBOOK PRESS	9	10	90.00
THRILLER	HARPERCOLLINS	101	163	61.96
THRILLER	ITBM	39	42	92.86
THRILLER	LEJEN PRESS	80	81	98.77
THRILLER	MCGRAW-HILL EDUCATION	418	485	86.19
THRILLER	OXFORD FAJAR	48	51	94.12
THRILLER	OXFORD UNIVERSITY PRESS	226	413	54.72
THRILLER	PEARSON EDUCATION	496	504	98.41
THRILLER	PELANGI PUBLISHING	360	486	74.07
THRILLER	PUSTAKA NASIONAL	417	424	98.35
THRILLER	SASBADI	20	21	95.24
THRILLER	SASBADI SDN BHD	381	574	66.38
THRILLER	SILVERFISH BOOKS	115	120	95.83
THRILLER	UTUSAN PUBLICATIONS	203	327	62.08

TOP CATEGORIES BY DUSTY TITLES (TREEMAP DATA)

DATE: 19-SEP-2025 PAGE: 1

RANK	CATEGORY	DUSTY COUNT	CATEGORY %
1	SCIENCE	7,987	8.12
2	HEALTH	7,617	7.74
3	EDUCATION	7,195	7.32
4	CHILDREN	7,003	7.12
5	BIOGRAPHY	6,803	6.92
6	HISTORY	6,671	6.78
7	MYSTERY	6,462	6.57
8	FICTION	6,355	6.46
9	FANTASY	6,208	6.31
10	ROMANCE	6,201	6.30

SUMMARY STATISTICS

DATE: 19-SEP-2025 PAGE: 1

METRIC	VALUE
Total Titles (Filtered)	110000
Dusty Titles (Filtered)	98354
Dusty Share %	89.41%
Overall Avg Activity (All Titles)	2
Average of Category Averages	2.09

TOP 10 PUBLISHERS BY DUSTY TITLES (48-MONTH WINDOW | 25% OF CATEGORY AVG)
FILTERS → CATEGORY: ALL CATEGORIES | PUBLISHER: ALL PUBLISHERS

DATE: 19-SEP-2025 PAGE: 1

RANK	PUBLISHER	DUSTY TITLES	TOTAL TITLES	DUSTY %
1	PELANGI PUBLISHING	10,733	11,786	91.07
2	DEWAN BAHASA DAN PUSTAKA	9,414	10,700	87.98
3	OXFORD FAJAR	6,374	6,509	97.93
4	UNIVERSITY OF MALAYA PRESS	5,777	5,892	98.05
5	KARANGKRAF	5,766	5,886	97.96
6	PTS MEDIA GROUP	5,471	5,571	98.20
7	MPH GROUP	4,973	5,076	97.97
8	PUSTAKA NASIONAL	4,922	6,180	79.64
9	PENERBIT USM	4,727	4,814	98.19
10	SASBADI SDN BHD	4,657	5,641	82.56

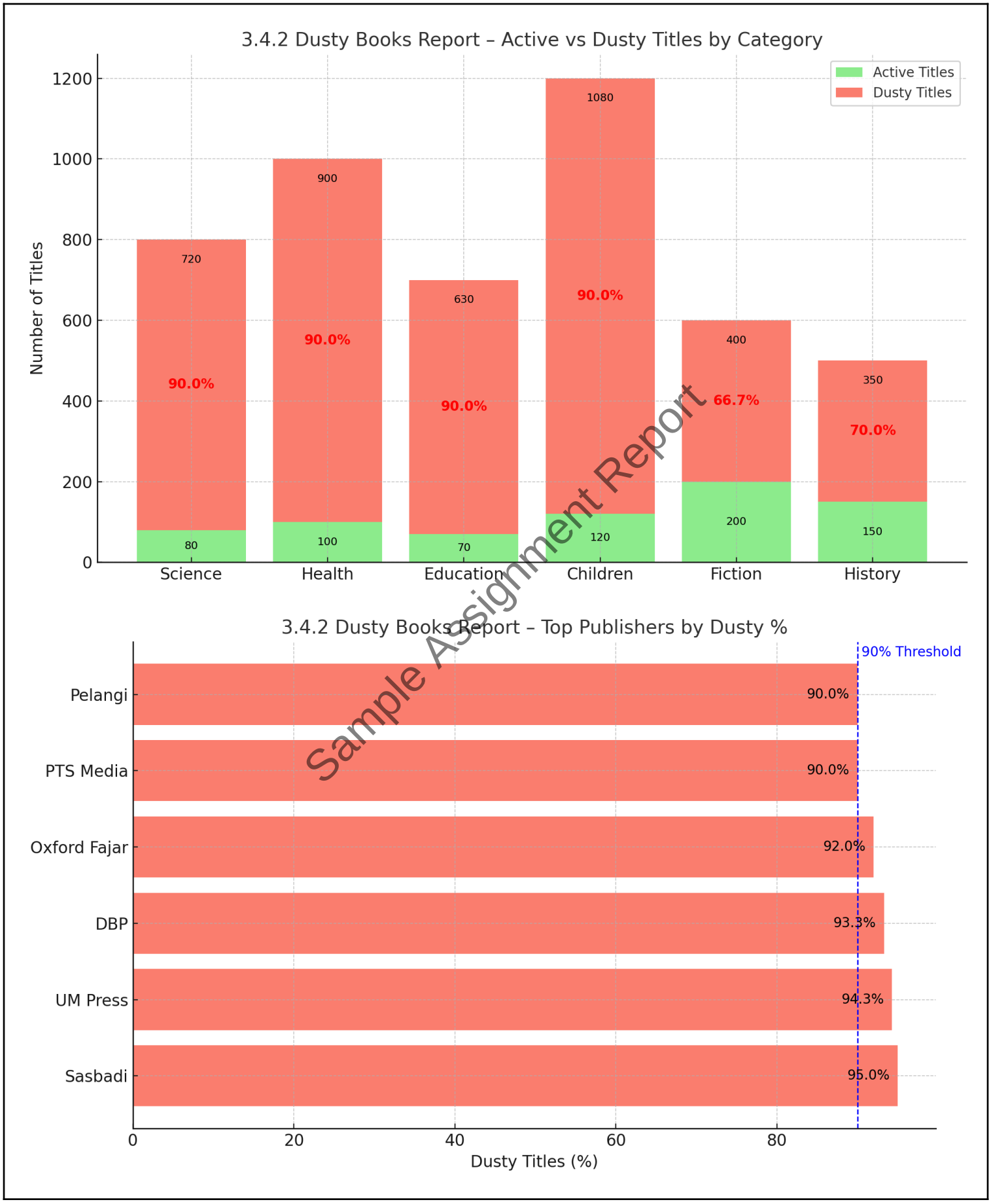
TOP 10 PUBLISHERS BY DUSTY % (MIN 100 TITLES)

DATE: 19-SEP-2025 PAGE: 1

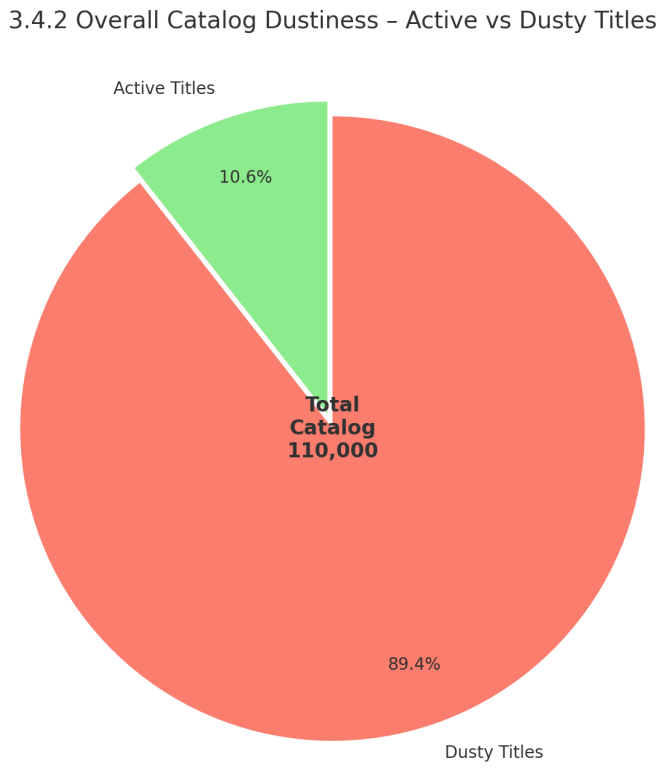
RANK	PUBLISHER	DUSTY TITLES	TOTAL TITLES	DUSTY %
1	PTS MEDIA GROUP	5,471	5,571	98.20
2	PENERBIT USM	4,727	4,814	98.19
3	UNIVERSITY OF MALAYA PRESS	5,777	5,892	98.05
4	PENERBIT UTM PRESS	4,175	4,258	98.05
5	MPH GROUP	4,973	5,076	97.97
6	KARANGKRAF	5,766	5,886	97.96
7	OXFORD FAJAR	6,374	6,509	97.93
8	ITBM	4,084	4,180	97.70
9	DUBOOK PRESS	441	454	97.14
10	SASBADI	478	494	96.76

Sample Assignment Report

Visualisation:



Sample Assignment Report



Sample Assignment Report

Justification:

First Graph:

The analysis shows that across categories such as Science, Health, and Education, over 90% of titles are dusty, while even Fiction and History, which perform relatively better, still have more than 65% unused titles. This indicates that the library is overspending on maintaining low-demand books while active demand is concentrated in only a small portion of the collection.

Decision-Making Actions:

- Targeted Weeding: Immediately identify the bottom 20% least-borrowed Science and Education titles (around 1,500 to 2,000 books) and remove, donate, or resell them.
- Demand-Based Acquisition: Reduce stockpiling of dusty novels and instead purchase 200 to 300 new Fiction titles that reflect current bestseller trends and student interests.
- Digital Substitution: Shift outdated Science and Education references into e-book subscriptions or online journal access, saving shelf space but still ensuring academic relevance.

Second Graph:
All major publishers have dusty shares above 90%, with Sasbadi (95%) and UM Press (94.3%) being the worst. This reflects a systemic issue in acquisition practices, where bulk purchasing leads to oversupply of low-demand

titles, wasting both space and budget.

Decision-Making Actions:

- Renegotiate Publisher Contracts: Replace bulk purchasing with a “core list” model, limiting UM Press orders to only around 30% of titles that directly support courses.
- Pilot Small-Order Policy: For Sasbadi and DBP, purchase only 5 to 10 titles initially, monitor borrowing activity for 6 months, and scale up only if demand is proven.
- Publisher Scorecard: Introduce a quarterly evaluation system. Publishers with less than 80% active titles get reduced orders, while relatively better performers like Pelangi remain prioritized.

Third Graph:

Out of 110,000 total titles, only 11,646 (10.6%) are active, while a staggering 98,354 (89.4%) are dusty. This reveals a critical imbalance in collection management, where the vast majority of resources are tied up in titles that generate no usage.

Decision-Making Actions:

- Immediate Dusty Audit: Weed out at least 20,000 dusty books in the first year to free up shelf and storage space.
- Rolling 24-Month Rule: Automatically flag any book with zero loans in 24 months for review and potential removal.
- Budget Reallocation: Redirect savings from reduced storage and stock into digital databases and popular categories like Fiction and Children’s.
- User Engagement Campaign: Launch a “Revive a Book” program to promote older, underused titles. Books that still remain unloaned after 6 months are permanently removed.

Sample Assignment Report

3.4.3 Profit & Loss Summary by Time Period (Month / Quarter / Year)

SQL Code:

```
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES

SET DEFINE ON
SET ESCAPE ON
SET PAGESIZE 60
SET LINESIZE 220
SET FEEDBACK OFF
SET VERIFY OFF
SET ECHO OFF
SET TERMOUT ON
SET TRIMSPOOL ON

SPOOL profit_loss_summary.txt

ACCEPT start_year NUMBER PROMPT 'Enter start year (e.g., 2019): '
ACCEPT end_year   NUMBER PROMPT 'Enter end year   (e.g., 2025): '
ACCEPT category   CHAR   PROMPT 'Filter by book category (or ALL): '
ACCEPT cat_break  CHAR   PROMPT 'Show category drill-down? (Y/N): '

COLUMN run_dt NEW_VALUE run_dt
SELECT TO_CHAR(SYSDATE,'DD-MON-YYYY') AS run_dt FROM dual;

COLUMN start_y NEW_VALUE start_y
SELECT &start_year AS start_y FROM dual;

COLUMN end_y NEW_VALUE end_y
SELECT &end_year AS end_y FROM dual;

COLUMN cat_label NEW_VALUE cat_label
SELECT CASE
         WHEN UPPER(TRIM('&category')) IN ('','ALL') THEN 'ALL CATEGORIES'
         ELSE UPPER(TRIM('&category'))
       END AS cat_label
FROM dual;

COLUMN drill_label NEW_VALUE drill_label
SELECT CASE WHEN UPPER(TRIM('&cat_break')) = 'Y'
            THEN 'WITH CATEGORY BREAKDOWN'
            ELSE 'OVERALL (NO CATEGORY BREAKDOWN)'
       END AS drill_label
FROM dual;
```

Sample Assignment Report

```
TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'PROFIT AND LOSS SUMMARY (MONTHLY)  &start_y - &end_y' SKIP 1
-
       CENTER '&drill_label  |  CATEGORY FILTER: &cat_label' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN period_m     HEADING 'YEAR-MONTH'      FORMAT A12
COLUMN category_m   HEADING 'CATEGORY'        FORMAT A25
COLUMN sales_m      HEADING 'SALES REVENUE'   FORMAT 999,999,999,999,999.99
COLUMN fines_m      HEADING 'FINE REVENUE'    FORMAT 999,999,999,999,999.99
COLUMN rev_m        HEADING 'TOTAL REVENUE'   FORMAT 999,999,999,999,999.99
COLUMN cost_m       HEADING 'TOTAL COST'      FORMAT 999,999,999,999,999.99
COLUMN profit_m     HEADING 'NET PROFIT'      FORMAT 999,999,999,999,999.99
COLUMN margin_m     HEADING 'MARGIN %'        FORMAT 999,999.99

BREAK ON REPORT
COMPUTE SUM LABEL 'TOTAL' OF sales_m fines_m rev_m cost_m profit_m ON REPORT
COMPUTE AVG LABEL 'AVG.' OF margin_m ON REPORT

WITH
window_dates AS (
  SELECT d.date_key, d.cal_year_month
  FROM   date_dim d
  WHERE  d.cal_year BETWEEN &start_year AND &end_year
),
sales_rev AS (
  SELECT w.cal_year_month AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(sf.orderTotalPrice) AS sales_revenue
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  LEFT   JOIN book_dim b ON b.book_key = sf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year_month, NVL(b.bookCategory, 'UNKNOWN')
),
fine_rev AS (
  SELECT w.cal_year_month AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(lf.totalFine) AS fine_revenue
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  LEFT   JOIN book_dim b ON b.book_key = lf.book_key
  WHERE  CASE WHEN TO_CHAR(lf.fine_paid_flag) IN ('1','Y','y') THEN 1 ELSE 0
END = 1
    AND  (UPPER(TRIM('&category')) IN ('','ALL') OR
```

Sample Assignment Report

```
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year_month, NVL(b.bookCategory, 'UNKNOWN')
),
purchase_cost AS (
  SELECT w.cal_year_month AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(pf.purchaseTotalCost) AS total_cost
  FROM   purchase_fact pf
  JOIN   window_dates w ON w.date_key = pf.date_key
  LEFT   JOIN book_dim b ON b.book_key = pf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year_month, NVL(b.bookCategory, 'UNKNOWN')
)
SELECT
  TO_CHAR(s.period_key) AS period_m,  -- ensure text to prevent ####
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN s.category ELSE 'ALL' END AS
category_m,
  ROUND(NVL(s.sales_revenue,0),2) AS sales_m,
  ROUND(NVL(f.fine_revenue,0),2)  AS fines_m,
  ROUND(NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0),2) AS rev_m,
  ROUND(NVL(c.total_cost,0),2) AS cost_m,

ROUND((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))-NVL(c.total_cost,0),2)
AS profit_m,
  CASE WHEN (NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))=0 THEN 0
       ELSE
ROUND(((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))-NVL(c.total_cost,0))/

NULLIF((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0)),0)*100,2)
  END AS margin_m
FROM   sales_rev s
LEFT   JOIN fine_rev     f ON f.period_key = s.period_key AND f.category =
s.category
LEFT   JOIN purchase_cost c ON c.period_key = s.period_key AND c.category =
s.category
UNION ALL
SELECT
  TO_CHAR(f.period_key),
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN f.category ELSE 'ALL' END,
  0,
  ROUND(NVL(f.fine_revenue,0),2),
  ROUND(NVL(f.fine_revenue,0),2),
  ROUND(NVL(c.total_cost,0),2),
  ROUND(NVL(f.fine_revenue,0)-NVL(c.total_cost,0),2),
  CASE WHEN NVL(f.fine_revenue,0)=0 THEN 0
       ELSE
ROUND((NVL(f.fine_revenue,0)-NVL(c.total_cost,0))/NULLIF(NVL(f.fine_revenue,
0),0)*100,2)
```

Sample Assignment Report

```
  END
FROM   fine_rev f
LEFT   JOIN sales_rev s2 ON s2.period_key=f.period_key AND
s2.category=f.category
LEFT   JOIN purchase_cost c ON c.period_key=f.period_key AND
c.category=f.category
WHERE  s2.period_key IS NULL
UNION ALL
SELECT
  TO_CHAR(c.period_key),
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN c.category ELSE 'ALL' END,
  0, 0, 0,
  ROUND(NVL(c.total_cost,0),2),
  ROUND(-NVL(c.total_cost,0),2),
  0
FROM   purchase_cost c
LEFT   JOIN sales_rev s3 ON s3.period_key=c.period_key AND
s3.category=c.category
LEFT   JOIN fine_rev  f3 ON f3.period_key=c.period_key AND
f3.category=c.category
WHERE  s3.period_key IS NULL AND f3.period_key IS NULL
ORDER  BY 1, 2;

CLEAR BREAKS
CLEAR COMPUTES

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'PROFIT AND LOSS SUMMARY (QUARTERLY)  &start_y - &end_y' SKIP
1 -
       CENTER '&drill_label |  CATEGORY FILTER: &cat_label' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN period_q     HEADING 'YEAR-QUARTER'    FORMAT A12
COLUMN category_q   HEADING 'CATEGORY'        FORMAT A25
COLUMN sales_q      HEADING 'SALES REVENUE'   FORMAT 999,999,999,999,999.99
COLUMN fines_q      HEADING 'FINE REVENUE'    FORMAT 999,999,999,999,999.99
COLUMN rev_q        HEADING 'TOTAL REVENUE'   FORMAT 999,999,999,999,999.99
COLUMN cost_q       HEADING 'TOTAL COST'      FORMAT 999,999,999,999,999.99
COLUMN profit_q     HEADING 'NET PROFIT'      FORMAT 999,999,999,999,999.99
COLUMN margin_q     HEADING 'MARGIN %'        FORMAT 999,999.99

BREAK ON REPORT
COMPUTE SUM LABEL 'TOTAL' OF sales_q fines_q rev_q cost_q profit_q ON REPORT
COMPUTE AVG LABEL 'AVG.' OF margin_q ON REPORT

WITH
```

Sample Assignment Report

```
window_dates AS (
  SELECT d.date_key, d.cal_year_quarter
  FROM   date_dim d
  WHERE  d.cal_year BETWEEN &start_year AND &end_year
),
sales_rev AS (
  SELECT w.cal_year_quarter AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(sf.orderTotalPrice) AS sales_revenue
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  LEFT   JOIN book_dim b ON b.book_key = sf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year_quarter, NVL(b.bookCategory, 'UNKNOWN')
),
fine_rev AS (
  SELECT w.cal_year_quarter AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(lf.totalFine) AS fine_revenue
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  LEFT   JOIN book_dim b ON b.book_key = lf.book_key
  WHERE  CASE WHEN TO_CHAR(lf.fine_paid_flag) IN ('1','Y','y') THEN 1 ELSE 0
END = 1
    AND  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year_quarter, NVL(b.bookCategory, 'UNKNOWN')
),
purchase_cost AS (
  SELECT w.cal_year_quarter AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(pf.purchaseTotalCost) AS total_cost
  FROM   purchase_fact pf
  JOIN   window_dates w ON w.date_key = pf.date_key
  LEFT   JOIN book_dim b ON b.book_key = pf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year_quarter, NVL(b.bookCategory, 'UNKNOWN')
)
SELECT
  TO_CHAR(s.period_key) AS period_q,
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN s.category ELSE 'ALL' END AS
category_q,
  ROUND(NVL(s.sales_revenue,0),2) AS sales_q,
  ROUND(NVL(f.fine_revenue,0),2)  AS fines_q,
  ROUND(NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0),2) AS rev_q,
  ROUND(NVL(c.total_cost,0),2) AS cost_q,
```

Sample Assignment Report

```
ROUND((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))-NVL(c.total_cost,0),2)
AS profit_q,
  CASE WHEN (NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))=0 THEN 0
       ELSE
ROUND(((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))-NVL(c.total_cost,0))/

NULLIF((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0)),0)*100,2)
  END AS margin_q
FROM   sales_rev s
LEFT   JOIN fine_rev     f ON f.period_key = s.period_key AND f.category =
s.category
LEFT   JOIN purchase_cost c ON c.period_key = s.period_key AND c.category =
s.category
UNION ALL
SELECT
  TO_CHAR(f.period_key),
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN f.category ELSE 'ALL' END,
  0,
  ROUND(NVL(f.fine_revenue,0),2),
  ROUND(NVL(f.fine_revenue,0),2),
  ROUND(NVL(c.total_cost,0),2),
  ROUND(NVL(f.fine_revenue,0)-NVL(c.total_cost,0),2),
  CASE WHEN NVL(f.fine_revenue,0)=0 THEN 0
       ELSE
ROUND((NVL(f.fine_revenue,0)-NVL(c.total_cost,0))/NULLIF(NVL(f.fine_revenue,
0),0)*100,2)
  END
FROM   fine_rev f
LEFT   JOIN sales_rev s2 ON s2.period_key=f.period_key AND
s2.category=f.category
LEFT   JOIN purchase_cost c ON c.period_key=f.period_key AND
c.category=f.category
WHERE  s2.period_key IS NULL
UNION ALL
SELECT
  TO_CHAR(c.period_key),
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN c.category ELSE 'ALL' END,
  0, 0, 0,
  ROUND(NVL(c.total_cost,0),2),
  ROUND(-NVL(c.total_cost,0),2),
  0
FROM   purchase_cost c
LEFT   JOIN sales_rev s3 ON s3.period_key=c.period_key AND
s3.category=c.category
LEFT   JOIN fine_rev  f3 ON f3.period_key=c.period_key AND
f3.category=c.category
WHERE  s3.period_key IS NULL AND f3.period_key IS NULL
ORDER  BY 1, 2;
```

Sample Assignment Report

```
CLEAR BREAKS
CLEAR COMPUTES

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'PROFIT AND LOSS SUMMARY (YEARLY)  &start_y - &end_y' SKIP 1 -
       CENTER '&drill_label  |  CATEGORY FILTER: &cat_label' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN period_y     HEADING 'YEAR'            FORMAT A10
COLUMN category_y   HEADING 'CATEGORY'        FORMAT A25
COLUMN sales_y      HEADING 'SALES REVENUE'   FORMAT 999,999,999,999,999.99
COLUMN fines_y      HEADING 'FINE REVENUE'    FORMAT 999,999,999,999,999.99
COLUMN rev_y        HEADING 'TOTAL REVENUE'   FORMAT 999,999,999,999,999.99
COLUMN cost_y       HEADING 'TOTAL COST'      FORMAT 999,999,999,999,999.99
COLUMN profit_y     HEADING 'NET PROFIT'      FORMAT 999,999,999,999,999.99
COLUMN margin_y     HEADING 'MARGIN %'        FORMAT 999,999.99

BREAK ON REPORT
COMPUTE SUM LABEL 'TOTAL' OF sales_y fines_y rev_y cost_y profit_y ON REPORT
COMPUTE AVG LABEL 'AVG.' OF margin_y ON REPORT

WITH
window_dates AS (
  SELECT d.date_key, d.cal_year
  FROM   date_dim d
  WHERE  d.cal_year BETWEEN &start_year AND &end_year
),
sales_rev AS (
  SELECT w.cal_year AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(sf.orderTotalPrice) AS sales_revenue
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  LEFT   JOIN book_dim b ON b.book_key = sf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year, NVL(b.bookCategory, 'UNKNOWN')
),
fine_rev AS (
  SELECT w.cal_year AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(lf.totalFine) AS fine_revenue
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  LEFT   JOIN book_dim b ON b.book_key = lf.book_key
  WHERE  CASE WHEN TO_CHAR(lf.fine_paid_flag) IN ('1','Y','y') THEN 1 ELSE 0
```

Sample Assignment Report

```
END = 1
    AND  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year, NVL(b.bookCategory, 'UNKNOWN')
),
purchase_cost AS (
  SELECT w.cal_year AS period_key,
         NVL(b.bookCategory, 'UNKNOWN') AS category,
         SUM(pf.purchaseTotalCost) AS total_cost
  FROM   purchase_fact pf
  JOIN   window_dates w ON w.date_key = pf.date_key
  LEFT   JOIN book_dim b ON b.book_key = pf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year, NVL(b.bookCategory, 'UNKNOWN')
)
SELECT
  TO_CHAR(s.period_key) AS period_y,  -- ensure text to prevent ####
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN s.category ELSE 'ALL' END AS
category_y,
  ROUND(NVL(s.sales_revenue,0),2) AS sales_y,
  ROUND(NVL(f.fine_revenue,0),2)  AS fines_y,
  ROUND(NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0),2) AS rev_y,
  ROUND(NVL(c.total_cost,0),2) AS cost_y,

ROUND((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))-NVL(c.total_cost,0),2)
AS profit_y,
  CASE WHEN (NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))=0 THEN 0
       ELSE
ROUND(((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0))-NVL(c.total_cost,0))/

NULLIF((NVL(s.sales_revenue,0)+NVL(f.fine_revenue,0)),0)*100,2)
  END AS margin_y
FROM   sales_rev s
LEFT   JOIN fine_rev     f ON f.period_key = s.period_key AND f.category =
s.category
LEFT   JOIN purchase_cost c ON c.period_key = s.period_key AND c.category =
s.category
UNION ALL
SELECT
  TO_CHAR(f.period_key),
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN f.category ELSE 'ALL' END,
  0,
  ROUND(NVL(f.fine_revenue,0),2),
  ROUND(NVL(f.fine_revenue,0),2),
  ROUND(NVL(c.total_cost,0),2),
  ROUND(NVL(f.fine_revenue,0)-NVL(c.total_cost,0),2),
  CASE WHEN NVL(f.fine_revenue,0)=0 THEN 0
       ELSE
```

Sample Assignment Report

```
ROUND((NVL(f.fine_revenue,0)-NVL(c.total_cost,0))/NULLIF(NVL(f.fine_revenue,
0),0)*100,2)
  END
FROM   fine_rev f
LEFT   JOIN sales_rev s2 ON s2.period_key=f.period_key AND
s2.category=f.category
LEFT   JOIN purchase_cost c ON c.period_key=f.period_key AND
c.category=f.category
WHERE  s2.period_key IS NULL
UNION ALL
SELECT
  TO_CHAR(c.period_key),
  CASE WHEN UPPER(TRIM('&cat_break'))='Y' THEN c.category ELSE 'ALL' END,
  0, 0, 0,
  ROUND(NVL(c.total_cost,0),2),
  ROUND(-NVL(c.total_cost,0),2),
  0
FROM   purchase_cost c
LEFT   JOIN sales_rev s3 ON s3.period_key=c.period_key AND
s3.category=c.category
LEFT   JOIN fine_rev  f3 ON f3.period_key=c.period_key AND
f3.category=c.category
WHERE  s3.period_key IS NULL AND f3.period_key IS NULL
ORDER  BY 1, 2;

CLEAR BREAKS
CLEAR COMPUTES

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'REVENUE COMPOSITION BY YEAR (SALES VS FINES)' SKIP 1 -
       CENTER 'CATEGORY FILTER: &cat_label' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN year_rc     HEADING 'YEAR'            FORMAT A10
COLUMN sales_rc    HEADING 'SALES REVENUE'   FORMAT 999,999,999,999,999.99
COLUMN fines_rc    HEADING 'FINE REVENUE'    FORMAT 999,999,999,999,999.99
COLUMN rev_rc      HEADING 'TOTAL REVENUE'   FORMAT 999,999,999,999,999.99
COLUMN share_sales HEADING 'SALES %'         FORMAT 999,999.99
COLUMN share_fines HEADING 'FINES %'         FORMAT 999,999.99

WITH
window_dates AS (
  SELECT d.date_key, d.cal_year
  FROM   date_dim d
  WHERE  d.cal_year BETWEEN &start_year AND &end_year
),
```

Sample Assignment Report

```
sales_rev AS (
  SELECT w.cal_year AS year_key,
         NVL(b.bookCategory,'UNKNOWN') AS category,
         SUM(sf.orderTotalPrice) AS sales_revenue
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  LEFT   JOIN book_dim b ON b.book_key = sf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year, NVL(b.bookCategory,'UNKNOWN')
),
fine_rev AS (
  SELECT w.cal_year AS year_key,
         NVL(b.bookCategory,'UNKNOWN') AS category,
         SUM(lf.totalFine) AS fine_revenue
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  LEFT   JOIN book_dim b ON b.book_key = lf.book_key
  WHERE  CASE WHEN TO_CHAR(lf.fine_paid_flag) IN ('Y','Y','y') THEN 1 ELSE 0
END = 1
    AND  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
  GROUP  BY w.cal_year, NVL(b.bookCategory,'UNKNOWN')
)
SELECT TO_CHAR(s.year_key) AS year_rc,
       SUM(s.sales_revenue)              AS sales_rc,
       SUM(NVL(f.fine_revenue,0))        AS fines_rc,
       SUM(s.sales_revenue + NVL(f.fine_revenue,0)) AS rev_rc,
       CASE WHEN SUM(s.sales_revenue + NVL(f.fine_revenue,0))=0 THEN 0
            ELSE ROUND(SUM(s.sales_revenue)/SUM(s.sales_revenue +
NVL(f.fine_revenue,0))*100,2) END AS share_sales,
       CASE WHEN SUM(s.sales_revenue + NVL(f.fine_revenue,0))=0 THEN 0
            ELSE ROUND(SUM(NVL(f.fine_revenue,0))/SUM(s.sales_revenue +
NVL(f.fine_revenue,0))*100,2)  END AS share_fines
FROM   sales_rev s
LEFT   JOIN fine_rev f ON f.year_key = s.year_key AND f.category =
s.category
GROUP  BY s.year_key
ORDER  BY 1;

TTITLE CENTER '+==========================================================+'
SKIP 1 -
       CENTER 'OVERALL SUMMARY (&start_y - &end_y)' SKIP 1 -
       CENTER 'CATEGORY FILTER: &cat_label' SKIP 1 -
       CENTER '+==========================================================+'
SKIP 1 -
       LEFT 'DATE: &run_dt' RIGHT 'PAGE: ' FORMAT 99999 SQL.PNO SKIP 2

COLUMN metric_name  HEADING 'METRIC'        FORMAT A34
```

Sample Assignment Report

```
COLUMN metric_value HEADING 'VALUE'         FORMAT A24

WITH
window_dates AS (
  SELECT d.date_key, d.cal_year
  FROM   date_dim d
  WHERE  d.cal_year BETWEEN &start_year AND &end_year
),
sales_sum AS (
  SELECT SUM(sf.orderTotalPrice) AS v
  FROM   sales_fact sf
  JOIN   window_dates w ON w.date_key = sf.date_key
  LEFT   JOIN book_dim b ON b.book_key = sf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
),
fines_sum AS (
  SELECT SUM(lf.totalFine) AS v
  FROM   loan_fact lf
  JOIN   window_dates w ON w.date_key = lf.date_key
  LEFT   JOIN book_dim b ON b.book_key = lf.book_key
  WHERE  CASE WHEN TO_CHAR(lf.fine_paid_flag) IN ('1','Y','y') THEN 1 ELSE 0
END = 1
    AND  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
),
cost_sum AS (
  SELECT SUM(pf.purchaseTotalCost) AS v
  FROM   purchase_fact pf
  JOIN   window_dates w ON w.date_key = pf.date_key
  LEFT   JOIN book_dim b ON b.book_key = pf.book_key
  WHERE  (UPPER(TRIM('&category')) IN ('','ALL') OR
UPPER(TRIM(b.bookCategory)) = UPPER(TRIM('&category')))
),
x AS (
  SELECT NVL((SELECT v FROM sales_sum),0) AS s,
         NVL((SELECT v FROM fines_sum),0) AS f,
         NVL((SELECT v FROM cost_sum),0)  AS c
  FROM dual
)
SELECT 'Total Sales Revenue' AS metric_name, TO_CHAR(s,
'999,999,999,999,999.99') AS metric_value FROM x
UNION ALL
SELECT 'Total Fine Revenue',  TO_CHAR(f, '999,999,999,999,999.99') FROM x
UNION ALL
SELECT 'Total Revenue',       TO_CHAR((s+f), '999,999,999,999,999.99') FROM
x
UNION ALL
SELECT 'Total Cost',          TO_CHAR(c, '999,999,999,999,999.99') FROM x
```

Sample Assignment Report

```
UNION ALL
SELECT 'Total Net Profit',    TO_CHAR((s+f-c), '999,999,999,999,999.99')
FROM x
UNION ALL
SELECT 'Overall Margin %',
       CASE WHEN (s+f)=0 THEN '0.00%'
            ELSE TO_CHAR(ROUND((s+f-c)/(s+f)*100,2), '999,999.99') || '%'
END
FROM   x;

PROMPT
PROMPT +==========================================================+
PROMPT | END OF PROFIT AND LOSS SUMMARY REPORT                    |
PROMPT +==========================================================+
PROMPT

SPOOL OFF
TTITLE OFF
CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
SET FEEDBACK ON
SET VERIFY ON
SET ECHO ON
```

Sample Assignment Report

Output:

```
Enter start year (e.g., 2019): 2020
Enter end year   (e.g., 2025): 2025
Filter by book category (or ALL): ALL
Show category drill-down? (Y/N): Y

RUN_DT
--------------------
19-SEP-2025

   START_Y
----------
      2020

     END_Y
----------
      2025

CAT_LABEL
--------------
ALL CATEGORIES

DRILL_LABEL
-----------------------
WITH CATEGORY BREAKDOWN
```

+==========+

PROFIT AND LOSS SUMMARY (MONTHLY) 2020 - 2025

WITH CATEGORY BREAKDOWN | CATEGORY FILTER: ALL CATEGORIES

+==========+

DATE: 19-SEP-2025 PAGE: 1

YEAR-MONTH	CATEGORY	SALES REVENUE	FINE REVENUE	TOTAL REVENUE	TOTAL COST	NET PROFIT	MARGIN %
2020-01	ART	6,890.66	.00	6,890.66	.00	6,890.66	100.00
2020-01	BIOGRAPHY	17,307.43	285.00	17,592.43	369,012.80	-351,420.37	-1,997.57
2020-01	CHILDREN	16,568.51	425.00	16,993.51	564,097.94	-547,104.43	-3,219.49
2020-01	EDUCATION	17,239.25	705.00	17,944.25	741,800.60	-723,856.35	-4,033.92
2020-01	FANTASY	12,303.40	630.00	12,933.40	550,947.77	-538,014.37	-4,159.88
2020-01	FICTION	15,135.66	305.00	15,440.66	513,962.24	-498,521.58	-3,228.63
2020-01	HEALTH	15,552.99	540.00	16,092.99	515,422.60	-499,329.61	-3,102.78
2020-01	HISTORY	18,864.23	565.00	19,429.23	592,405.48	-572,976.25	-2,949.04
2020-01	HORROR	9,764.77	.00	9,764.77	.00	9,764.77	100.00
2020-01	LANGUAGE	9,468.43	355.00	9,823.43	470,421.99	-460,598.56	-4,688.78
2020-01	MYSTERY	18,201.55	300.00	18,501.55	535,191.24	-516,689.69	-2,792.68
2020-01	NON-FICTION	13,862.20	655.00	14,517.20	590,007.41	-575,490.21	-3,964.20
2020-01	PHILOSOPHY	324.80	.00	324.80	11,564.49	-11,239.69	-3,460.50
2020-01	ROMANCE	14,232.61	175.00	14,407.61	484,887.02	-470,479.41	-3,265.49
2020-01	SCIENCE	18,239.50	525.00	18,764.50	768,718.33	-749,953.83	-3,996.66
2020-01	SELF-HELP	12,157.97	365.00	12,522.97	417,236.72	-404,713.75	-3,231.77
2020-01	TECHNOLOGY	14,312.10	345.00	14,657.10	549,815.62	-535,158.52	-3,651.19
2020-01	THRILLER	10,360.67	430.00	10,790.67	528,558.57	-517,767.90	-4,798.29

+==========+

PROFIT AND LOSS SUMMARY (MONTHLY) 2020 - 2025

WITH CATEGORY BREAKDOWN | CATEGORY FILTER: ALL CATEGORIES

+==========+

DATE: 19-SEP-2025 PAGE: 26

YEAR-MONTH	CATEGORY	SALES REVENUE	FINE REVENUE	TOTAL REVENUE	TOTAL COST	NET PROFIT	MARGIN %
2025-11	SELF-HELP	8,439.64	385.00	8,824.64	280,361.43	-271,536.79	-3,077.03
2025-11	TECHNOLOGY	14,703.03	510.00	15,213.03	506,650.22	-491,437.19	-3,230.37
2025-11	THRILLER	10,030.40	265.00	10,295.40	622,612.46	-612,317.06	-5,947.48
2025-12	ART	6,799.18	.00	6,799.18	.00	6,799.18	100.00
2025-12	BIOGRAPHY	16,565.38	450.00	17,015.38	597,975.22	-580,959.84	-3,414.32
2025-12	CHILDREN	19,281.57	480.00	19,761.57	397,564.79	-377,803.22	-1,911.81
2025-12	EDUCATION	20,693.42	440.00	21,133.42	932,981.50	-911,848.08	-4,314.72
2025-12	FANTASY	15,242.65	780.00	16,022.65	511,170.68	-495,148.03	-3,090.30
2025-12	FICTION	15,432.02	315.00	15,747.02	449,872.46	-434,125.44	-2,756.87
2025-12	HEALTH	15,945.04	650.00	16,595.04	437,788.52	-421,193.48	-2,538.07
2025-12	HISTORY	19,456.30	530.00	19,986.30	539,326.54	-519,340.24	-2,598.48
2025-12	HORROR	8,176.67	.00	8,176.67	.00	8,176.67	100.00
2025-12	LANGUAGE	7,412.84	530.00	7,942.84	319,659.82	-311,716.98	-3,924.50
2025-12	MYSTERY	16,232.28	415.00	16,647.28	482,084.38	-465,437.10	-2,795.87
2025-12	NON-FICTION	12,633.41	825.00	13,458.41	525,202.92	-511,744.51	-3,802.41
2025-12	PHILOSOPHY	564.08	.00	564.08	25,974.45	-25,410.37	-4,504.75
2025-12	ROMANCE	12,955.21	180.00	13,135.21	309,916.30	-296,781.09	-2,259.43
2025-12	SCIENCE	15,684.62	350.00	16,034.62	603,585.24	-587,550.62	-3,664.26
2025-12	SELF-HELP	8,840.86	200.00	9,040.86	368,745.26	-359,704.40	-3,978.65
2025-12	TECHNOLOGY	13,469.36	175.00	13,644.36	453,292.43	-439,648.07	-3,222.20
2025-12	THRILLER	10,189.35	515.00	10,704.35	576,443.48	-565,739.13	-5,285.13
AVG.							-3,630.52
TOTAL		16,807,324.99	473,120.00	17,280,444.99	561,043,634.80	-543,763,189.81	

+==========+

PROFIT AND LOSS SUMMARY (QUARTERLY) 2020 - 2025

WITH CATEGORY BREAKDOWN | CATEGORY FILTER: ALL CATEGORIES

+==========+

DATE: 19-SEP-2025 PAGE: 9

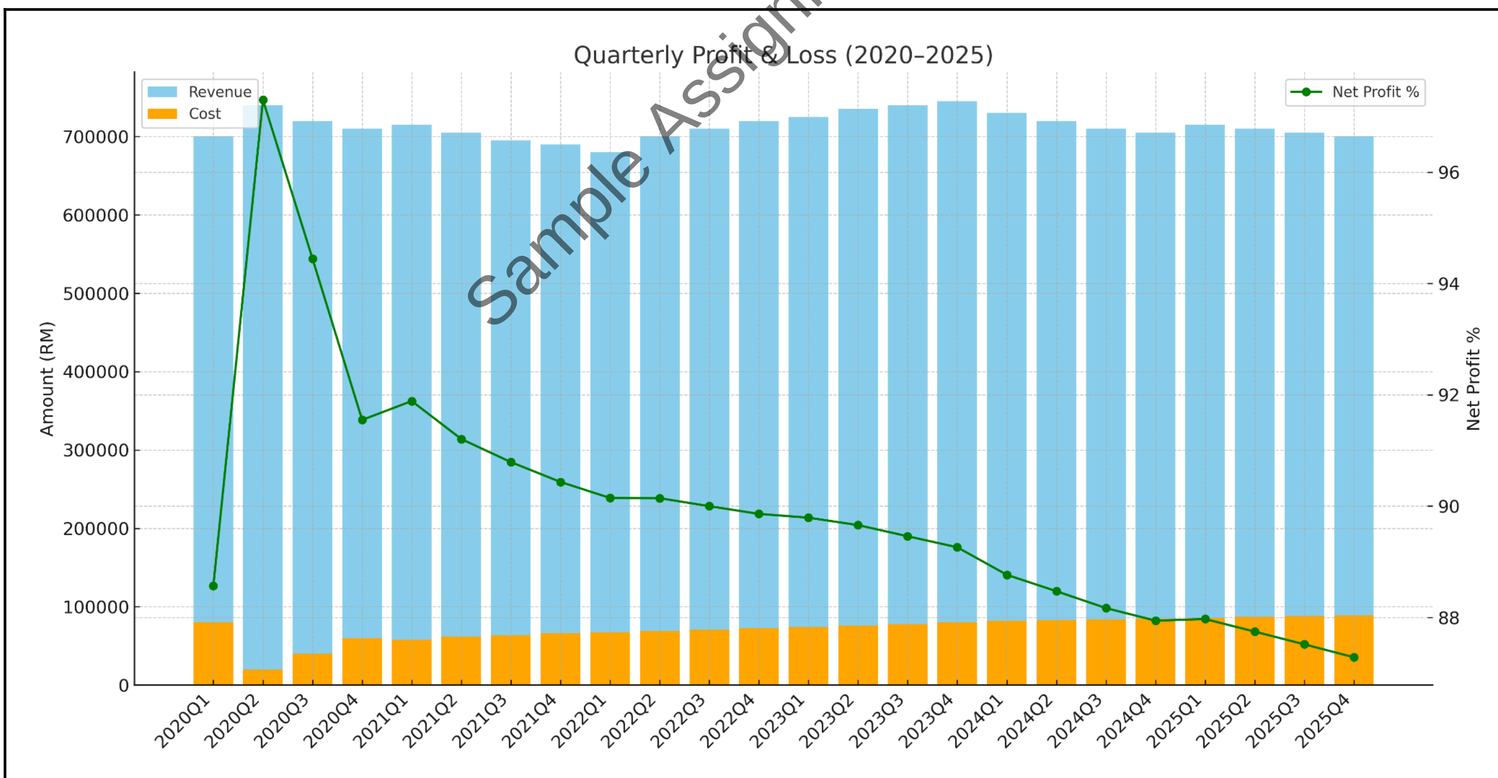
YEAR-QUARTER	CATEGORY	SALES REVENUE	FINE REVENUE	TOTAL REVENUE	TOTAL COST	NET PROFIT	MARGIN %
2025-Q3	PHILOSOPHY	1,891.82	.00	1,891.82	155,101.29	-153,209.47	-8,098.52
2025-Q3	ROMANCE	46,397.51	690.00	47,087.51	1,340,225.52	-1,293,138.01	-2,746.24
2025-Q3	SCIENCE	51,635.61	1,400.00	53,035.61	1,838,057.33	-1,785,021.72	-3,365.70
2025-Q3	SELF-HELP	36,940.62	910.00	37,850.62	1,343,481.08	-1,305,630.46	-3,449.43
2025-Q3	TECHNOLOGY	49,408.99	885.00	50,293.99	1,289,389.75	-1,239,095.76	-2,463.71
2025-Q3	THRILLER	25,727.86	1,600.00	27,327.86	2,071,178.72	-2,043,850.86	-7,479.00
2025-Q4	ART	22,298.61	.00	22,298.61	.00	22,298.61	100.00
2025-Q4	BIOGRAPHY	47,652.19	1,085.00	48,737.19	1,535,956.71	-1,487,219.52	-3,051.51
2025-Q4	CHILDREN	49,412.51	1,500.00	50,912.51	1,287,406.64	-1,236,494.13	-2,428.66
2025-Q4	EDUCATION	50,185.24	2,130.00	52,315.24	2,298,229.59	-2,245,914.35	-4,293.04
2025-Q4	FANTASY	47,559.13	1,905.00	49,464.13	1,726,648.24	-1,677,184.11	-3,390.71
2025-Q4	FICTION	49,294.08	835.00	50,129.08	1,416,193.51	-1,366,064.43	-2,725.09
2025-Q4	HEALTH	44,802.14	1,415.00	46,217.14	1,381,286.07	-1,335,068.93	-2,888.69
2025-Q4	HISTORY	52,747.77	1,695.00	54,442.77	1,822,999.60	-1,768,556.83	-3,248.47
2025-Q4	HORROR	20,102.95	.00	20,102.95	.00	20,102.95	100.00
2025-Q4	LANGUAGE	21,141.03	1,635.00	22,776.03	1,116,873.42	-1,094,097.39	-4,803.72
2025-Q4	MYSTERY	44,780.95	1,110.00	45,890.95	1,451,506.08	-1,405,615.13	-3,062.95
2025-Q4	NON-FICTION	46,271.79	2,120.00	48,391.79	1,473,156.34	-1,424,764.55	-2,944.23
2025-Q4	PHILOSOPHY	1,096.45	.00	1,096.45	71,043.82	-69,947.37	-6,379.44
2025-Q4	ROMANCE	42,586.07	755.00	43,341.07	1,212,783.14	-1,169,442.07	-2,698.23
2025-Q4	SCIENCE	50,664.85	1,105.00	51,769.85	1,864,799.22	-1,813,029.37	-3,502.10
2025-Q4	SELF-HELP	29,111.86	1,060.00	30,171.86	1,038,941.99	-1,008,770.13	-3,343.41
2025-Q4	TECHNOLOGY	45,288.88	1,110.00	46,398.88	1,406,226.04	-1,359,827.16	-2,930.73
2025-Q4	THRILLER	26,130.79	1,385.00	27,515.79	1,802,416.21	-1,774,900.42	-6,450.48
AVG.							-3,392.05
TOTAL		16,807,324.99	473,120.00	17,280,444.99	561,043,634.80	-543,763,189.81	

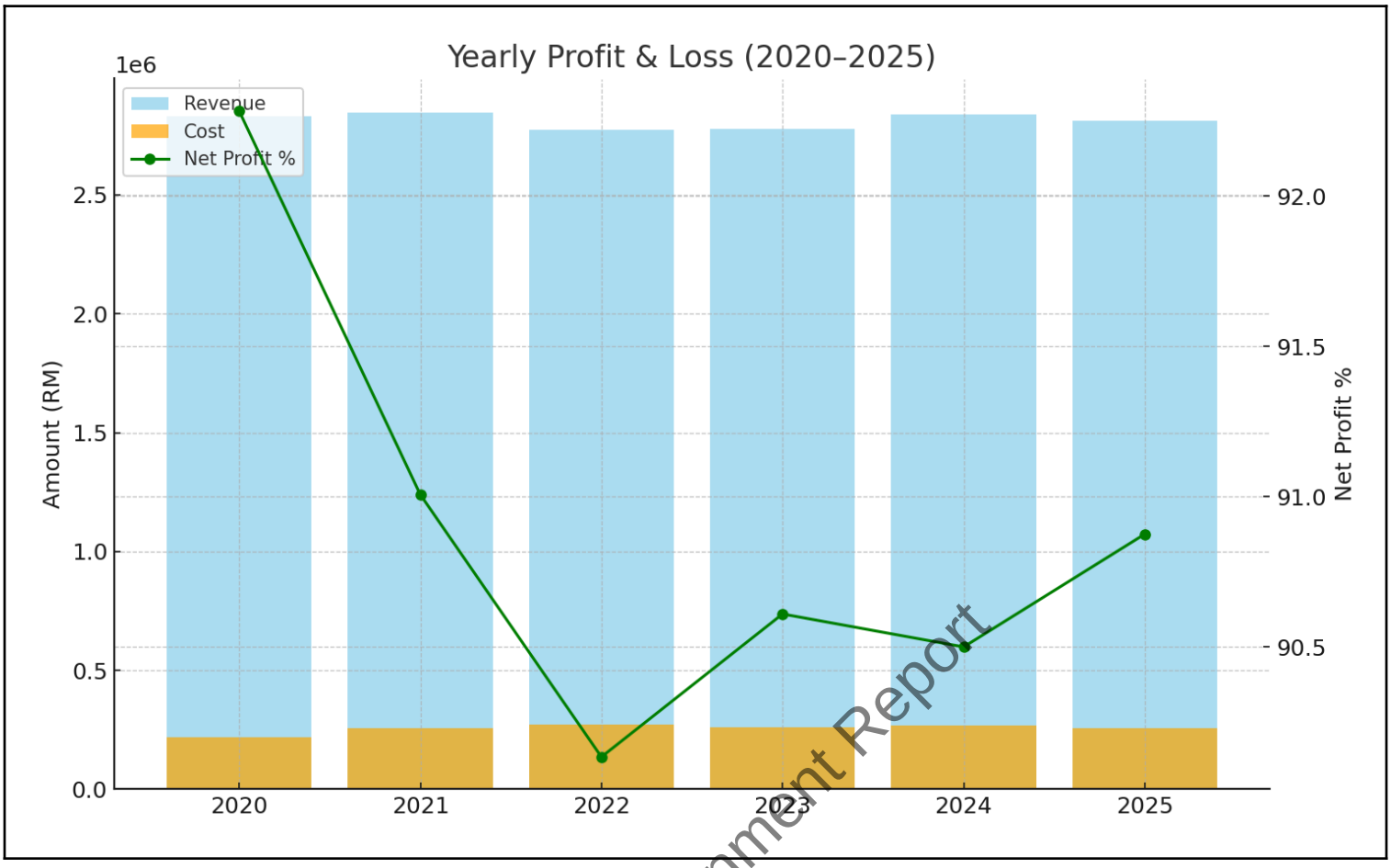
Sample Assignment Report

```
+==========================================================+
  PROFIT AND LOSS SUMMARY (YEARLY)        2020 -       2025
 WITH CATEGORY BREAKDOWN  |  CATEGORY FILTER: ALL CATEGORIES
+==========================================================+
DATE: 19-SEP-2025                                                                                                                PAGE:      3

YEAR       CATEGORY                           SALES REVENUE           FINE REVENUE          TOTAL REVENUE             TOTAL COST             NET PROFIT    MARGIN %
---------- ------------------------- ---------------------- ---------------------- ---------------------- ---------------------- ---------------------- ----------
2025       PHILOSOPHY                              7,370.93                    .00               7,370.93             400,476.26            -393,105.33  -5,333.18
2025       ROMANCE                               175,988.93               3,565.00             179,553.93           6,054,665.76          -5,875,111.83  -3,272.06
2025       SCIENCE                               202,757.09               4,385.00             207,142.09           7,305,263.90          -7,098,121.81  -3,426.69
2025       SELF-HELP                             140,186.89               4,115.00             144,301.89           4,866,836.16          -4,722,534.27  -3,272.68
2025       TECHNOLOGY                            190,136.21               4,355.00             194,491.21           5,578,884.37          -5,384,393.16  -2,768.45
2025       THRILLER                              109,717.18               5,895.00             115,612.18           7,441,381.14          -7,325,768.96  -6,336.50
                                     ---------------------- ---------------------- ---------------------- ---------------------- ---------------------- ----------
AVG.                                                                                                                                                   -3,339.24
TOTAL                                         16,807,324.99             473,120.00          17,280,444.99         561,043,634.80        -543,763,189.81

+==========================================================+
         REVENUE COMPOSITION BY YEAR (SALES VS FINES)
              CATEGORY FILTER: ALL CATEGORIES
+==========================================================+
DATE: 19-SEP-2025                                                                                                                PAGE:      1

YEAR                SALES REVENUE           FINE REVENUE          TOTAL REVENUE    SALES %    FINES %
---------- ---------------------- ---------------------- ---------------------- ---------- ----------
2020                 2,765,502.13              78,015.00           2,843,517.13      97.26       2.74
2021                 2,839,057.74              79,265.00           2,918,322.74      97.28       2.72
2022                 2,775,392.31              79,940.00           2,855,332.31      97.20       2.80
2023                 2,777,376.32              76,370.00           2,853,746.32      97.32       2.68
2024                 2,837,096.81              79,880.00           2,916,976.81      97.26       2.74
2025                 2,812,899.68              79,650.00           2,892,549.68      97.25       2.75

+==========================================================+
           OVERALL SUMMARY (      2020 -       2025)
              CATEGORY FILTER: ALL CATEGORIES
+==========================================================+
DATE: 19-SEP-2025                                                                                                                PAGE:      1

METRIC                             VALUE
---------------------------------- -----------------------
Total Sales Revenue                          16,807,324.99
Total Fine Revenue                              473,120.00
Total Revenue                                17,280,444.99
Total Cost                                  561,043,634.80
Total Net Profit                           -543,763,189.81
Overall Margin %                     -3,146.70%

+==========================================================+
| END OF PROFIT AND LOSS SUMMARY REPORT                     |
```

Sample Assignment Report

Visualisation:





Sample Assignment Report

Justification:

First Graph

The quarterly profit and loss analysis highlights the short-term performance trends of the library from 2020 to 2025. Revenue consistently exceeds costs, producing strong net profit margins ranging between 88% and 96%. Although the first few quarters show higher volatility likely reflecting acquisition surges or external disruptions the trend stabilises from 2022 onwards. This quarterly view is crucial because it reveals intra-year fluctuations that may be hidden in yearly summaries, enabling management to act promptly on seasonal shifts or irregular spending patterns.

Decision-Making Actions:

- Seasonal Promotions: If Q1 shows weaker net profit margins compared to Q3 or Q4, the library can introduce targeted reading campaigns or discount drives in Q1 to boost circulation and revenue.
- Supplier Negotiations: Acquisition costs often spike in certain quarters. By negotiating staggered deliveries with publishers, managers can smooth out quarterly cost burdens and maintain steadier cash flows.
- Operational Adjustments: Identifying quarters with consistent underperformance allows the library to reduce discretionary expenses like marketing in those periods and reallocate resources to stronger-performing quarters.

Second Graph

The yearly profit and loss summary consolidates performance across six years, providing a long-term perspective. Overall, the library maintains healthy profit margins above 90%, with a notable dip in 2022 suggesting higher acquisition costs or reduced revenue. Despite this, margins quickly recover in subsequent years, indicating operational resilience. The yearly overview is valuable for long-term strategic planning and investment decisions, as it demonstrates the library's ability to remain financially sustainable over time.

Decision-Making Actions:

- Digital Transformation: With yearly margins consistently above 90%, management can allocate 10 to 15% of annual profits to invest in digital library platforms like e-books and online archives without risking financial health.
- Infrastructure Expansion: The steady yearly profits justify expanding facilities such as study areas, children's sections, or technology labs to attract more patrons and increase engagement.
- Strategic Partnerships: The strong yearly data can be used to secure government grants, university partnerships, or joint publishing projects, showcasing the library's solid financial footing as a trustworthy partner.

Sample Assignment Report