# PROJECT

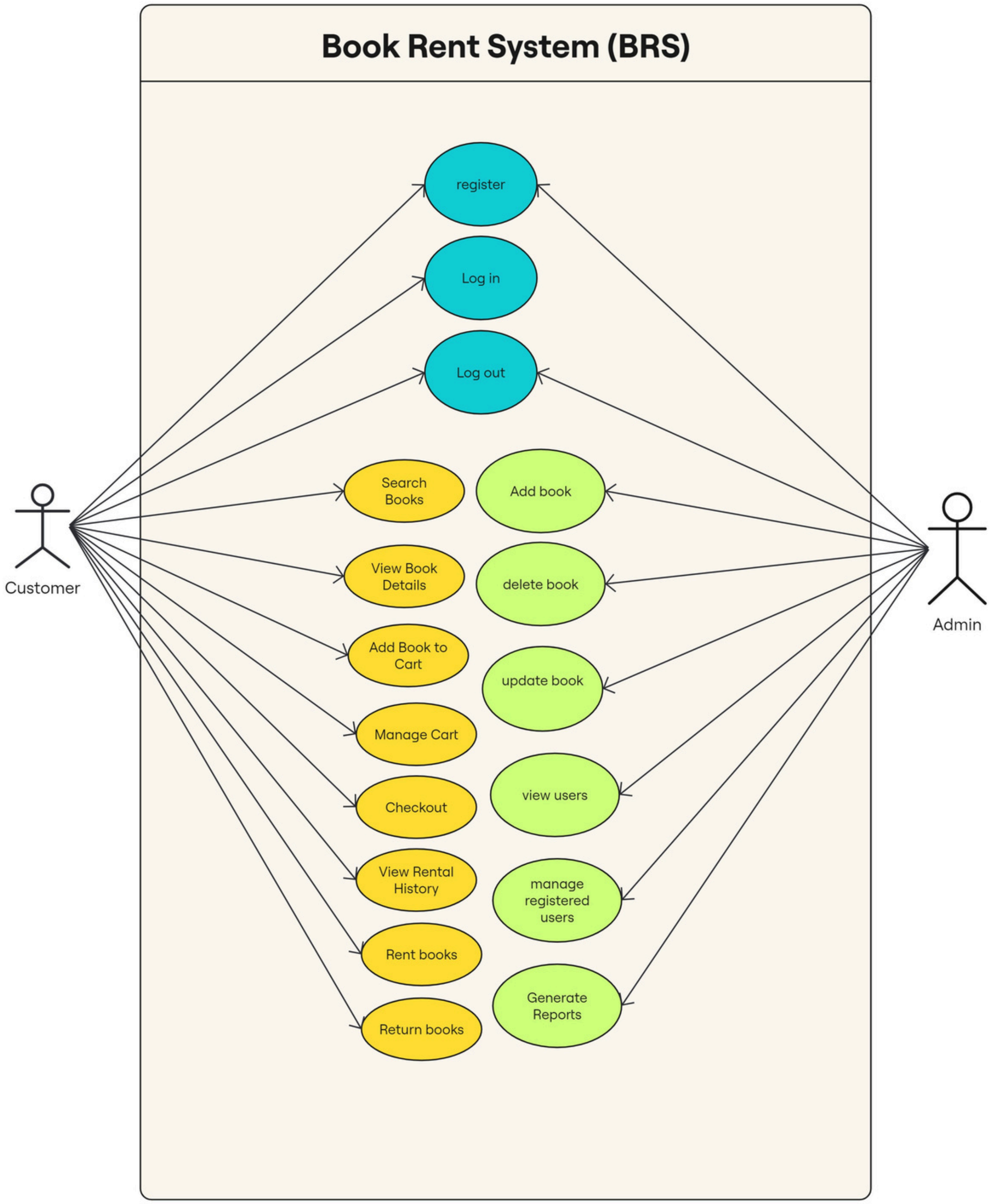
# DATABASE2

## Book Rent System (BRS)

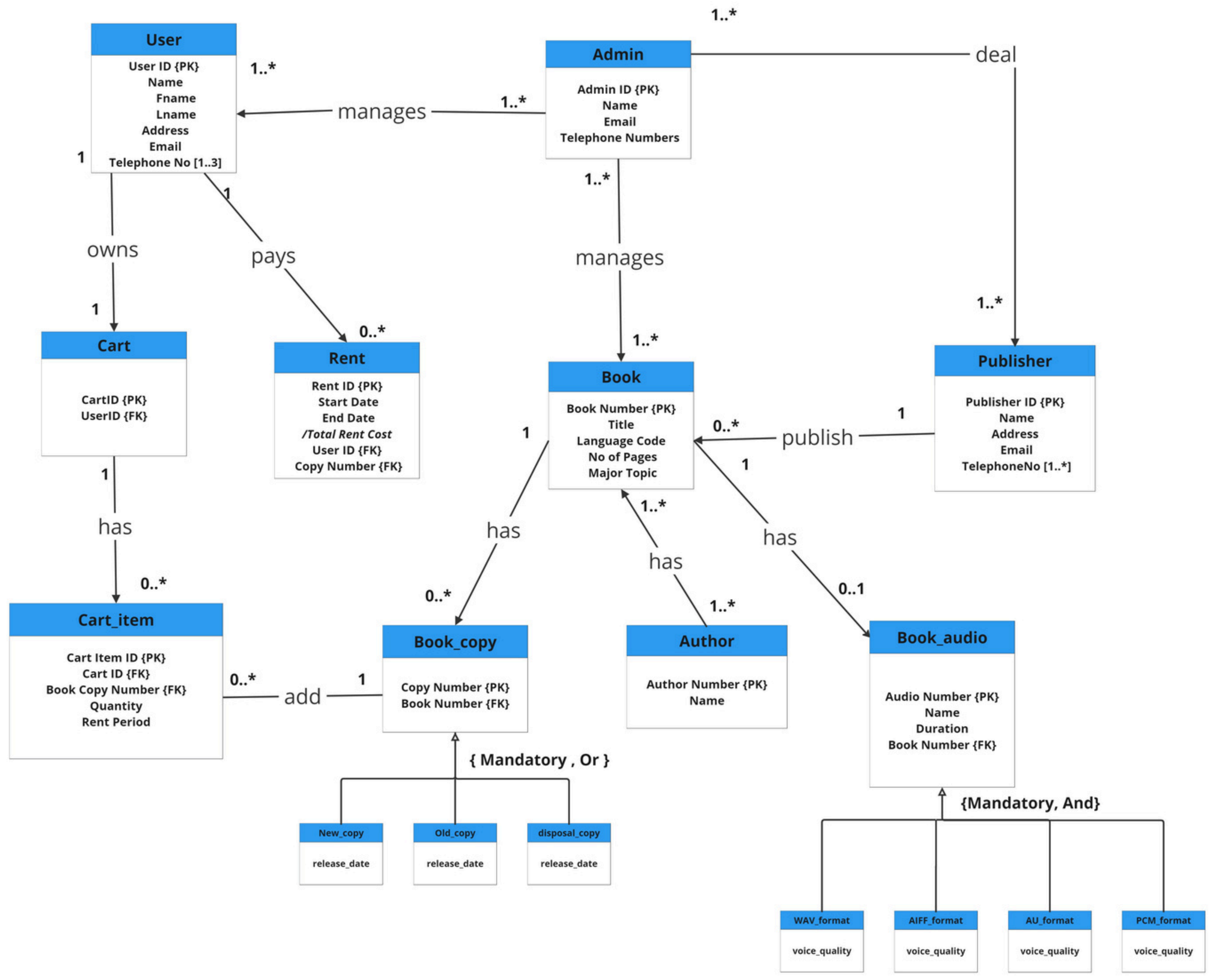
Prepared for :
**Mahmoud AL Saheb**

Prepared by :
**Ahmed Asafrah (211033)**
**Monjed Tamimi (211069)**

# use case model || phase 1



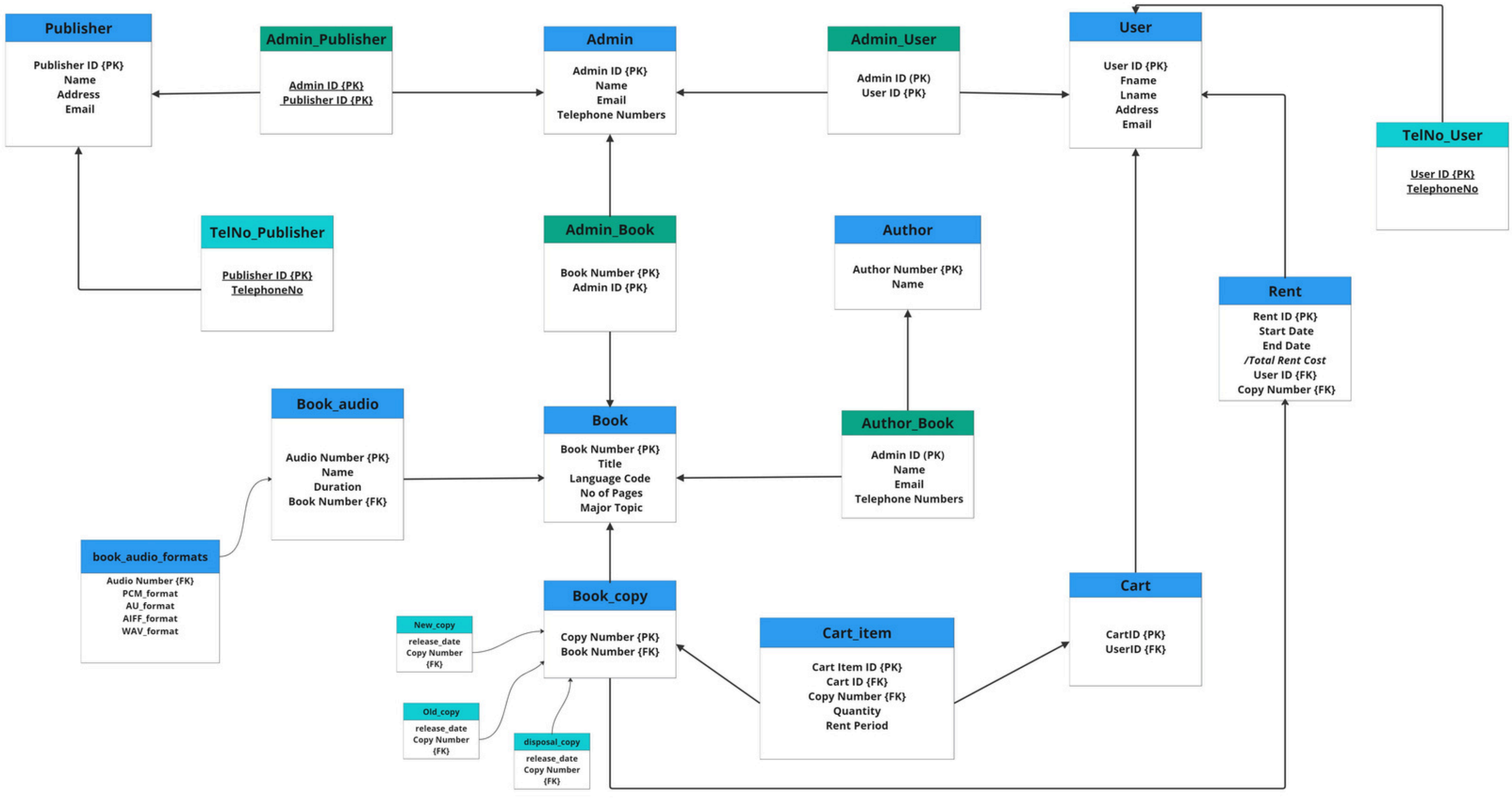
# conceptual model || phase 1



## FOR BETTER VIEW :

https://miro.com/app/board/uXjVKLx6YnA=/

# logical design || phase 2



## NORMALIZATION FOR MY PROJECT:

1. **First Normal Form (1NF)**: Ensures no duplicate columns in each table, breaks down complex tables into simpler ones based on distinct subjects, and assigns a unique primary key to each table for unique identification of records.

2. **Second Normal Form (2NF)**: Builds on 1NF by removing attributes that only partially depend on the primary key, creating separate tables for these attributes, and linking them back to the original table through foreign keys.

3. **Third Normal Form (3NF)**: Advances from 2NF by ensuring every attribute in a table directly depends on the primary key, not on other non-key attributes, thereby reducing transitive dependencies and further enhancing data integrity

# Table Creation:

```
CREATE TABLE "USER" (
    "USER_ID" NUMBER PRIMARY KEY,
    "NAME" VARCHAR2(255),
    "FNAME" VARCHAR2(255),
    "ADDRESS" VARCHAR2(255),
    "EMAIL" VARCHAR2(255) UNIQUE
);

CREATE TABLE "PUBLISHER" (
    "PUBLISHER_ID" NUMBER PRIMARY KEY,
    "NAME" VARCHAR2(255),
    "ADDRESS" VARCHAR2(255),
    "EMAIL" VARCHAR2(255)
);

CREATE TABLE "BOOK" (
    "BOOK_NUMBER" NUMBER PRIMARY KEY,
    "TITLE" VARCHAR2(255),
    "LANGUAGE_CODE" VARCHAR2(50),
    "NO_OF_PAGES" NUMBER,
    "MAJOR_TOPIC" VARCHAR2(255)
);

CREATE TABLE "BOOK_COPY" (
    "COPY_NUMBER" NUMBER PRIMARY KEY,
    "BOOK_NUMBER" NUMBER,
    "STATUS" VARCHAR2(50),
    "RELEASE_DATE" DATE,
    FOREIGN KEY ("BOOK_NUMBER") REFERENCES "BOOK"("BOOK_NUMBER")
);

CREATE TABLE "RENT" (
    "RENT_ID" NUMBER PRIMARY KEY,
    "START_DATE" DATE,
    "END_DATE" DATE,
    "TOTAL_RENT_COST" NUMBER(10, 2),
    "USER_ID" NUMBER,
    "COPY_NUMBER" NUMBER,
    FOREIGN KEY ("USER_ID") REFERENCES "USER"("USER_ID"),
    FOREIGN KEY ("COPY_NUMBER") REFERENCES "BOOK_COPY"("COPY_NUMBER")
);

CREATE TABLE "AUTHOR" (
    "AUTHOR_NUMBER" NUMBER PRIMARY KEY,
    "NAME" VARCHAR2(255)
);

CREATE TABLE "AUTHOR_BOOK" (
    "BOOK_NUMBER" NUMBER,
    "AUTHOR_NUMBER" NUMBER,
    PRIMARY KEY ("BOOK_NUMBER", "AUTHOR_NUMBER"),
    FOREIGN KEY ("BOOK_NUMBER") REFERENCES "BOOK"("BOOK_NUMBER"),
    FOREIGN KEY ("AUTHOR_NUMBER") REFERENCES "AUTHOR"("AUTHOR_NUMBER")
);

CREATE TABLE "CART" (
    "CART_ID" NUMBER PRIMARY KEY,
    "USER_ID" NUMBER,
    FOREIGN KEY ("USER_ID") REFERENCES "USER"("USER_ID")
);

CREATE TABLE "CART_ITEM" (
    "CART_ITEM_ID" NUMBER PRIMARY KEY,
    "CART_ID" NUMBER,
    "COPY_NUMBER" NUMBER,
    "QUANTITY" NUMBER,
    "RENT_PERIOD" NUMBER,
    FOREIGN KEY ("CART_ID") REFERENCES "CART"("CART_ID"),
    FOREIGN KEY ("COPY_NUMBER") REFERENCES "BOOK_COPY"("COPY_NUMBER")
);

CREATE TABLE "ADMIN" (
    "ADMIN_ID" NUMBER PRIMARY KEY,
    "NAME" VARCHAR2(255),
    "EMAIL" VARCHAR2(255),
    "TELEPHONE_NUMBERS" VARCHAR2(255)
);

CREATE TABLE "ADMIN_USER" (
    "ADMIN_ID" NUMBER,
    "USER_ID" NUMBER,
    PRIMARY KEY ("ADMIN_ID", "USER_ID"),
    FOREIGN KEY ("ADMIN_ID") REFERENCES "ADMIN"("ADMIN_ID"),
    FOREIGN KEY ("USER_ID") REFERENCES "USER"("USER_ID")
);

CREATE TABLE "TELNO_USER" (
    "USER_ID" NUMBER,
    "TELEPHONE_NO" VARCHAR2(20),
    PRIMARY KEY ("USER_ID", "TELEPHONE_NO"),
    FOREIGN KEY ("USER_ID") REFERENCES "USER"("USER_ID")
);

CREATE TABLE "TELNO_PUBLISHER" (
    "PUBLISHER_ID" NUMBER,
    "TELEPHONE_NO" VARCHAR2(20),
    PRIMARY KEY ("PUBLISHER_ID", "TELEPHONE_NO"),
    FOREIGN KEY ("PUBLISHER_ID") REFERENCES
"PUBLISHER"("PUBLISHER_ID")
);
```

```
CREATE TABLE "BOOK_AUDIO" (
    "AUDIO_NUMBER" NUMBER PRIMARY KEY,
    "NAME" VARCHAR2(255),
    "DURATION" NUMBER,
    "BOOK_NUMBER" NUMBER,
    FOREIGN KEY ("BOOK_NUMBER") REFERENCES "BOOK"("BOOK_NUMBER")
);

CREATE TABLE "BOOK_AUDIO_FORMATS" (
    "AUDIO_NUMBER" NUMBER,
    "PCM_FORMAT" CHAR(1),
    "AU_FORMAT" CHAR(1),
    "AIFF_FORMAT" CHAR(1),
    "WAV_FORMAT" CHAR(1),
    PRIMARY KEY ("AUDIO_NUMBER"),
    FOREIGN KEY ("AUDIO_NUMBER") REFERENCES
"BOOK_AUDIO"("AUDIO_NUMBER")
);

CREATE TABLE "ADMIN_PUBLISHER" (
    "ADMIN_ID" NUMBER,
    "PUBLISHER_ID" NUMBER,
    CONSTRAINT "PK_ADMIN_PUBLISHER" PRIMARY KEY ("ADMIN_ID",
"PUBLISHER_ID"),
    CONSTRAINT "FK_ADMIN_PUBLISHER_ADMIN_ID" FOREIGN KEY ("ADMIN_ID")
REFERENCES "ADMIN"("ADMIN_ID"),
    CONSTRAINT "FK_ADMIN_PUBLISHER
```

# Views Creation:

```
CREATE VIEW "VIEW_BOOKS_COPIES_STATUS" AS
SELECT b."BOOK_NUMBER", b."TITLE", bc."COPY_NUMBER", bc."STATUS"
FROM "BOOK" b
JOIN "BOOK_COPY" bc ON b."BOOK_NUMBER" = bc."BOOK_NUMBER";


CREATE VIEW "VIEW_ADMIN_BOOKS" AS
SELECT a."ADMIN_ID", a."NAME", b."BOOK_NUMBER", b."TITLE"
FROM "ADMIN" a
JOIN "ADMIN_BOOK" ab ON a."ADMIN_ID" = ab."ADMIN_ID"
JOIN "BOOK" b ON ab."BOOK_NUMBER" = b."BOOK_NUMBER";


CREATE VIEW "VIEW_ACTIVE_RENTS" AS
SELECT u."USER_ID", u."NAME", r."RENT_ID", r."START_DATE",
r."END_DATE", r."TOTAL_RENT_COST"
FROM "USER" u
JOIN "RENT" r ON u."USER_ID" = r."USER_ID"
```

```
WHERE r."END_DATE" >= SYSDATE;


CREATE VIEW "VIEW_AUTHOR_BOOKS" AS
SELECT a."AUTHOR_NUMBER", a."NAME" AS Author_Name, b."BOOK_NUMBER",
b."TITLE" AS Book_Title
FROM "AUTHOR" a
JOIN "AUTHOR_BOOK" ab ON a."AUTHOR_NUMBER" = ab."AUTHOR_NUMBER"
JOIN "BOOK" b ON ab."BOOK_NUMBER" = b."BOOK_NUMBER";


CREATE VIEW "VIEW_PUBLISHER_BOOKS" AS
SELECT p."PUBLISHER_ID", p."NAME" AS Publisher_Name, b."BOOK_NUMBER",
b."TITLE" AS Book_Title
FROM "PUBLISHER" p
JOIN "BOOK" b ON p."PUBLISHER_ID" = b."PUBLISHER_ID";
```

# Transactions:

### -- Transaction to Register a New User --

```
BEGIN
    INSERT INTO "USER" ("USER_ID", "NAME", "FNAME", "ADDRESS",
"EMAIL")
    VALUES (user_seq.NEXTVAL, 'John Doe', 'John', '1234 Main St',
'john.doe@example.com');

    INSERT INTO "TELNO_USER" ("USER_ID", "TELEPHONE_NO")
    VALUES (user_seq.CURRVAL, '123-456-7890');

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

### -- Transaction to Rent a Book --

```
BEGIN
    INSERT INTO "RENT" ("RENT_ID", "USER_ID", "COPY_NUMBER",
"START_DATE", "END_DATE", "TOTAL_RENT_COST")
    VALUES (rent_seq.NEXTVAL, 1, 101, SYSDATE, SYSDATE + 7, 19.99);

    UPDATE "BOOK_COPY"
    SET "STATUS" = 'rented'
    WHERE "COPY_NUMBER" = 101;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

## -- Transaction to Return a Book --

```
BEGIN
    UPDATE "RENT"
    SET "END_DATE" = SYSDATE
    WHERE "RENT_ID" = 1;

    UPDATE "BOOK_COPY"
    SET "STATUS" = 'available'
    WHERE "COPY_NUMBER" = 101;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

## -- Transaction to Add a New Book --

```
BEGIN
    INSERT INTO "BOOK" ("BOOK_NUMBER", "TITLE", "LANGUAGE_CODE",
"NO_OF_PAGES", "MAJOR_TOPIC")
    VALUES (book_seq.NEXTVAL, 'New Oracle Insights', 'EN', 300,
'Database Management');

    INSERT INTO "BOOK_COPY" ("COPY_NUMBER", "BOOK_NUMBER", "STATUS",
"RELEASE_DATE")
    VALUES (copy_seq.NEXTVAL, book_seq.CURRVAL, 'new', SYSDATE);

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

## -- Transaction to Assign an Admin to a Book --

```
BEGIN
    INSERT INTO "ADMIN_BOOK" ("ADMIN_ID", "BOOK_NUMBER")
    VALUES (1, 200);  -- Assuming ADMIN_ID 1 and BOOK_NUMBER 200 are
valid and existing

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

## Reports Creation :

### -- Report on Active Rents –

```
SELECT u."USER_ID", u."NAME" AS "User_Name", b."TITLE" AS
"Book_Title",
       r."START_DATE", r."END_DATE", r."TOTAL_RENT_COST"
FROM "RENT" r
JOIN "USER" u ON r."USER_ID" = u."USER_ID"
JOIN "BOOK_COPY" bc ON r."COPY_NUMBER" = bc."COPY_NUMBER"
JOIN "BOOK" b ON bc."BOOK_NUMBER" = b."BOOK_NUMBER"
WHERE r."END_DATE" >= SYSDATE;
```

### -- Report on Book Copies by Status –

```
SELECT "STATUS", COUNT(*) AS "Number_of_Copies"
FROM "BOOK_COPY"
GROUP BY "STATUS";

-- Report on Overdue Rents
SELECT u."USER_ID", u."NAME" AS "User_Name", b."TITLE" AS
"Book_Title",
       r."RENT_ID", r."START_DATE", r."END_DATE", r."TOTAL_RENT_COST"
FROM "RENT" r
JOIN "USER" u ON r."USER_ID" = u."USER_ID"
JOIN "BOOK_COPY" bc ON r."COPY_NUMBER" = bc."COPY_NUMBER"
JOIN "BOOK" b ON bc."BOOK_NUMBER" = b."BOOK_NUMBER"
WHERE r."END_DATE" < SYSDATE;
```

-- Report on Book Rentals Per User --

```
SELECT u."USER_ID", u."NAME" AS "User_Name", COUNT(r."RENT_ID") AS
"Total_Rents"
FROM "RENT" r
JOIN "USER" u ON r."USER_ID" = u."USER_ID"
GROUP BY u."USER_ID", u."NAME";
```

-- Report on Books Managed by Each Admin --

```
SELECT a."ADMIN_ID", a."NAME" AS "Admin_Name", b."BOOK_NUMBER",
b."TITLE" AS "Book_Title"
FROM "ADMIN_BOOK" ab
JOIN "ADMIN" a ON ab."ADMIN_ID" = a."ADMIN_ID"
JOIN "BOOK" b ON ab."BOOK_NUMBER" = b."BOOK_NUMBER";
```

# Index Creation

Index on User Email for faster lookup during user login and verification processes

```
CREATE INDEX "IDX_USER_EMAIL" ON "USER" ("EMAIL");
```

Indexes on Book for quick searches by Major Topic and Language Code

```
CREATE INDEX "IDX_BOOK_MAJOR_TOPIC" ON "BOOK" ("MAJOR_TOPIC");
CREATE INDEX "IDX_BOOK_LANGUAGE_CODE" ON "BOOK" ("LANGUAGE_CODE");
```

Index on Book Copy Status to enhance the performance of queries filtering by copy status

```
CREATE INDEX "IDX_BOOK_COPY_STATUS" ON "BOOK_COPY" ("STATUS");
```

Indexes on Rent for quick access to rental records by User ID and Copy Number

```
CREATE INDEX "IDX_RENT_USER_ID" ON "RENT" ("USER_ID");
CREATE INDEX "IDX_RENT_COPY_NUMBER" ON "RENT" ("COPY_NUMBER");
```

Composite index on Author_Book to facilitate efficient

```
CREATE INDEX "IDX_AUTHOR_BOOK" ON "AUTHOR_BOOK" ("BOOK_NUMBER",
"AUTHOR_NUMBER");
```