|  |  |
| --- | --- |
| Submitted By: | Gangula Karthik (223715Y) |
|  | Jay Krish Vijendra (222197Y) |
|  | Ng Jun Ming (220080D) |
|  | Seah Pin Shien (220273H) |
| PEM Group: | BA2202 |
| Tutor: | Liz Wang |
| Submitted on: | 18/Dec/2022 |

ITB421 Data STORAGE ADMINISTRATION

Assignment - PART 1

Physical Data Modelling

# Part 1: Defining Database Structure and Objects using Database Definition Language (DDL) for OraRental database (60 marks)

**Task 1: Develop DDL statements for creating the tables in the OraRental database**

|  |
| --- |
| **ORARENT\_ACTORS** |
| *-- DDL STATEMENTS FOR THE ORARENT ACTORS*  CREATE TABLE **ORARENT\_ACTORS**  ( ACTOR\_ID NUMBER(10),  STAGE\_NAME VARCHAR2(40) NOT NULL,  FIRST\_NAME VARCHAR2(25) NOT NULL,  LAST\_NAME VARCHAR2(25) NOT NULL,  BIRTH\_DATE DATE NOT NULL,  CONSTRAINT actors\_actor\_id\_pk PRIMARY KEY(ACTOR\_ID)  );  DESCRIBE ORARENT\_ACTORS; |
| **ORARENT\_CUSTOMERS** |
| *-- DDL STATEMENTS FOR THE ORARENT CUSTOMERS*  CREATE TABLE **ORARENT\_CUSTOMERS**  ( CUSTOMER\_ID NUMBER(10) NOT NULL,  LAST\_NAME VARCHAR2(25) NOT NULL,  FIRST\_NAME VARCHAR2(25) NOT NULL,  HOME\_PHONE VARCHAR2(12) NOT NULL,  ADDRESS VARCHAR2(100) NOT NULL,  CITY VARCHAR2(30) NOT NULL,  STATE VARCHAR2(2) NOT NULL,  EMAIL VARCHAR2(25),  CELL\_PHONE VARCHAR2(12),  CONSTRAINT customers\_customer\_id\_pk PRIMARY KEY(CUSTOMER\_ID)  );  DESCRIBE ORARENT\_CUSTOMERS;  **Calendar  Description automatically generated** |

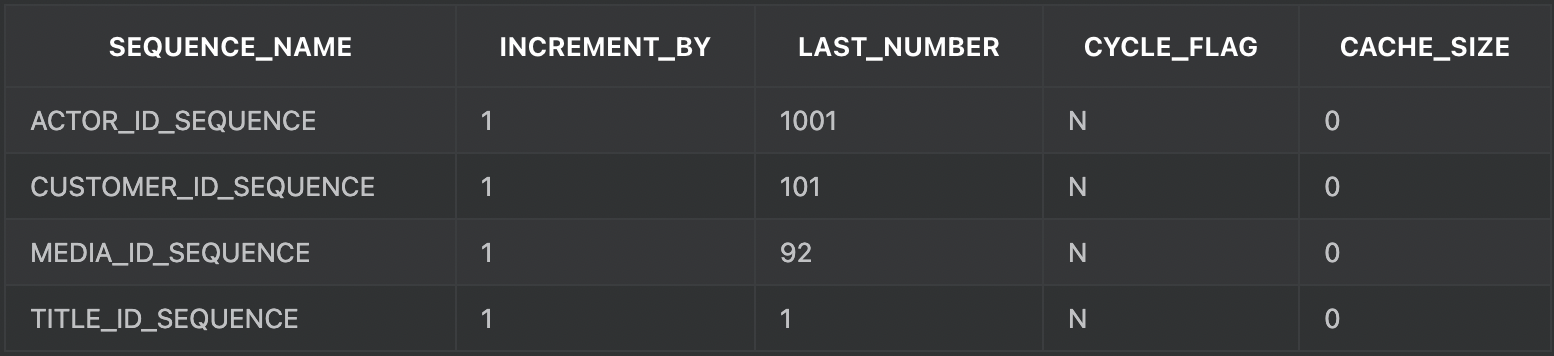
|  |
| --- |
| **ORARENT\_MEDIA** |
| *-- DDL STATEMENTS FOR THE ORARENT MEDIA*  CREATE TABLE **ORARENT\_MEDIA**  ( MEDIA\_ID NUMBER(10),  FORMAT VARCHAR2(3) NOT NULL,  TITLE\_ID NUMBER(10) NOT NULL,  CONSTRAINT media\_title\_id\_pk PRIMARY KEY(MEDIA\_ID),  CONSTRAINT media\_title\_id\_fk FOREIGN KEY (TITLE\_ID) REFERENCES ORARENT\_MOVIES(TITLE\_ID)  );  DESCRIBE ORARENT\_MEDIA;  **A screenshot of a computer  Description automatically generated with medium confidence** |
| **ORARENT\_MOVIES** |
| *-- DDL STATEMENTS FOR THE ORARENT MOVIES*  CREATE TABLE **ORARENT\_MOVIES**  ( TITLE\_ID NUMBER(10),  TITLE VARCHAR2(60) NOT NULL,  DESCRIPTION VARCHAR2(400) NOT NULL,  RATING VARCHAR2(4),  CATEGORY VARCHAR2(20),  RELEASE\_DATE DATE NOT NULL,  CONSTRAINT movies\_title\_id\_pk PRIMARY KEY(TITLE\_ID),  CONSTRAINT movies\_rating\_check CHECK(RATING IN ('G', 'PG', 'PG13', 'R')),  CONSTRAINT movies\_category\_check CHECK(CATEGORY IN ('DRAMA', 'COMEDY', 'ACTION', 'CHILD', 'SCIFI', 'DOCUMENTARY'))  );  DESCRIBE ORARENT\_MOVIES; |
| **ORARENT\_RENTAL\_HISTORY** |
| *-- DDL STATEMENTS FOR THE ORARENT RENTAL\_HISTORY*  CREATE TABLE **ORARENT\_RENTAL\_HISTORY**  ( MEDIA\_ID NUMBER(10),  RENTAL\_DATE DATE DEFAULT SYSDATE,  CUSTOMER\_ID NUMBER(10) NOT NULL,  RETURN\_DATE DATE,  CONSTRAINT rental\_hist\_id\_rent\_date\_pk PRIMARY KEY(MEDIA\_ID, RENTAL\_DATE),  CONSTRAINT rental\_hist\_media\_id\_fk FOREIGN KEY (MEDIA\_ID) REFERENCES ORARENT\_MEDIA(MEDIA\_ID),  CONSTRAINT rental\_hist\_customer\_id\_fk FOREIGN KEY (CUSTOMER\_ID) REFERENCES ORARENT\_CUSTOMERS(CUSTOMER\_ID)  );  DESCRIBE ORARENT\_RENTAL\_HISTORY; |
| **ORARENT\_STAR\_BILLINGS** |
| *-- DDL STATEMENTS FOR THE ORARENT STAR\_BILLINGS*  CREATE TABLE **ORARENT\_STAR\_BILLINGS**  ( ACTOR\_ID NUMBER(10),  TITLE\_ID NUMBER(10),  COMMENTS VARCHAR2(40),  CONSTRAINT st\_billings\_id\_rent\_date\_pk PRIMARY KEY(ACTOR\_ID, TITLE\_ID),  CONSTRAINT star\_billings\_actor\_id\_fk FOREIGN KEY (ACTOR\_ID) REFERENCES ORARENT\_ACTORS(ACTOR\_ID),  CONSTRAINT star\_billings\_title\_id\_fk FOREIGN KEY (TITLE\_ID) REFERENCES ORARENT\_MOVIES(TITLE\_ID)  );  DESCRIBE ORARENT\_STAR\_BILLINGS; |

**Task 2: Creating and Managing Constraints**

|  |
| --- |
| **ORARENT\_ACTORS** |
| Primary Key:    *-- SQL STATEMENT TO VERIFY IF PRIMARY KEY EXISTS*  SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  WHERE TABLE\_NAME LIKE 'ORARENT\_ACTORS' AND CONSTRAINT\_TYPE = 'P';  Not Null constraint:    SELECT TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH, NULLABLE  FROM ALL\_TAB\_COLUMNS  WHERE TABLE\_NAME LIKE 'ORARENT\_ACTORS';  *Note: "N" in the Nullable column indicates that the column cannot have a null value, while "Y" indicates that it can.* |
| **ORARENT\_CUSTOMERS** |
| Primary Key:    SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  where TABLE\_NAME like 'ORARENT\_CUSTOMERS' AND CONSTRAINT\_TYPE = 'P';  Not Null constraint:  Graphical user interface, application  Description automatically generated  SELECT TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH, NULLABLE  FROM ALL\_TAB\_COLUMNS  WHERE TABLE\_NAME LIKE 'ORARENT\_CUSTOMERS'; |
| **ORARENT\_MEDIA** |
| Primary Key:    SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  WHERE TABLE\_NAME LIKE 'ORARENT\_MEDIA' AND CONSTRAINT\_TYPE = 'P';  Foreign Key:    SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  Where TABLE\_NAME like 'ORARENT\_MEDIA' and CONSTRAINT\_TYPE = 'R';  Not Null constraint:    SELECT TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH, NULLABLE  FROM ALL\_TAB\_COLUMNS  WHERE TABLE\_NAME LIKE 'ORARENT\_MEDIA'; |
| **ORARENT\_MOVIES** |
| Primary Key:    SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  WHERE TABLE\_NAME LIKE 'ORARENT\_MOVIES' AND CONSTRAINT\_TYPE = 'P';  Not Null constraint:    SELECT TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH, NULLABLE  FROM ALL\_TAB\_COLUMNS  WHERE TABLE\_NAME LIKE 'ORARENT\_MOVIES';  Check constraints:    SELECT CONSTRAINT\_NAME, CONSTRAINT\_TYPE, SEARCH\_CONDITION  FROM ALL\_CONSTRAINTS  WHERE TABLE\_NAME = 'ORARENT\_MOVIES'  AND CONSTRAINT\_TYPE = 'C'; |
| **ORARENT\_RENTAL\_HISTORY** |
| Primary Key:    SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  WHERE TABLE\_NAME LIKE 'ORARENT\_RENTAL\_HISTORY' AND CONSTRAINT\_TYPE = 'P';  Foreign Key:    SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  WHERE TABLE\_NAME LIKE 'ORARENT\_RENTAL\_HISTORY' AND CONSTRAINT\_TYPE = 'R';  Not Null constraint:      SELECT TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH, NULLABLE  FROM ALL\_TAB\_COLUMNS  WHERE TABLE\_NAME LIKE 'ORARENT\_RENTAL\_HISTORY'; |
| **ORARENT\_STAR\_BILLINGS** |
| Primary Key:    SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  WHERE TABLE\_NAME LIKE 'ORARENT\_STAR\_BILLINGS' AND CONSTRAINT\_TYPE = 'P';  Foreign Key:    SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, STATUS  FROM USER\_CONSTRAINTS  WHERE TABLE\_NAME LIKE 'ORARENT\_STAR\_BILLINGS' AND CONSTRAINT\_TYPE = 'R';  Not Null constraint:    SELECT TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH, NULLABLE  FROM ALL\_TAB\_COLUMNS  WHERE TABLE\_NAME LIKE 'ORARENT\_STAR\_BILLINGS'; |

**Task 3: Working with Sequences (Indexes and Synonyms) & View**

1. Create the following sequences to be used for primary key values:



CREATE SEQUENCE **CUSTOMER\_ID\_SEQUENCE**

INCREMENT BY 1

START WITH 101

CACHE

NOCYCLE;

CREATE SEQUENCE **TITLE\_ID\_SEQUENCE**

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE **MEDIA\_ID\_SEQUENCE**

INCREMENT BY 1

START WITH 92

NOCACHE

NOCYCLE;

CREATE SEQUENCE **ACTOR\_ID\_SEQUENCE**

INCREMENT BY 1

START WITH 1001

NOCACHE

NOCYCLE;

*-- VERIFY THAT THE SEQUENCE WAS CREATED*

SELECT SEQUENCE\_NAME, INCREMENT\_BY, LAST\_NUMBER, CYCLE\_FLAG, CACHE\_SIZE

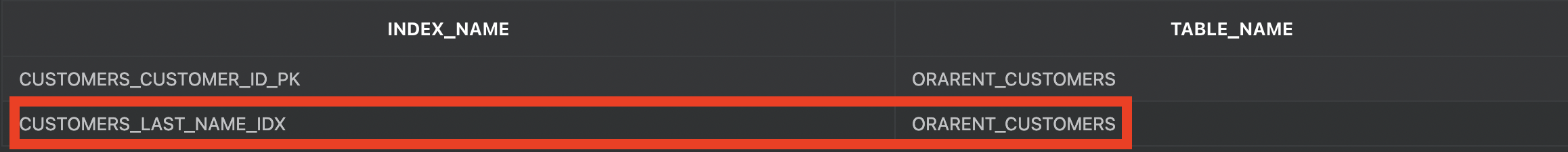
FROM USER\_SEQUENCES;

1. Add the data to the tables using INSERT statements.

|  |
| --- |
| **ORARENT\_ACTORS** |
| *-- ORARENT ACTORS INSERTS*  *-- ROW 1*  INSERT INTO ORARENT\_ACTORS(ACTOR\_ID,STAGE\_NAME,FIRST\_NAME,LAST\_NAME,BIRTH\_DATE)  VALUES (ACTOR\_ID\_SEQUENCE.NEXTVAL, 'Tom Hanks', 'Thomas', 'Hanks', '09-Jul-1956');  *-- ROW 2*  INSERT INTO ORARENT\_ACTORS(ACTOR\_ID,STAGE\_NAME,FIRST\_NAME,LAST\_NAME,BIRTH\_DATE)  VALUES (ACTOR\_ID\_SEQUENCE.NEXTVAL, 'Jim Carrey', 'James', 'Carrey', '17-Jan-1962');  *-- ROW 3*  INSERT INTO ORARENT\_ACTORS(ACTOR\_ID,STAGE\_NAME,FIRST\_NAME,LAST\_NAME,BIRTH\_DATE)  VALUES (ACTOR\_ID\_SEQUENCE.NEXTVAL, 'Keanu Reeves', 'Keanu', 'Reeves', '02-Sep-1964');  *-- ROW 4*  INSERT INTO ORARENT\_ACTORS(ACTOR\_ID,STAGE\_NAME,FIRST\_NAME,LAST\_NAME,BIRTH\_DATE)  VALUES (ACTOR\_ID\_SEQUENCE.NEXTVAL, 'Sam Worthington', 'Samuel', 'Worthington', '02-Aug-1976');  *-- ROW 5*  INSERT INTO ORARENT\_ACTORS(ACTOR\_ID,STAGE\_NAME,FIRST\_NAME,LAST\_NAME,BIRTH\_DATE)  VALUES (ACTOR\_ID\_SEQUENCE.NEXTVAL, 'Eddie Murphy', 'Edward', 'Murphy', '03-Apr-1961');  *-- ROW 6*  INSERT INTO ORARENT\_ACTORS(ACTOR\_ID,STAGE\_NAME,FIRST\_NAME,LAST\_NAME,BIRTH\_DATE)  VALUES (ACTOR\_ID\_SEQUENCE.NEXTVAL, 'Tom Cruise', 'Thomas', 'Cruise', '03-Jul-1962');  SELECT \* FROM ORARENT\_ACTORS; |
| **ORARENT\_CUSTOMERS** |
| *-- ORARENT CUSTOMERS INSERTS*  *-- ROW 1*  INSERT INTO ORARENT\_CUSTOMERS(CUSTOMER\_ID, LAST\_NAME, FIRST\_NAME, HOME\_PHONE, ADDRESS, CITY, STATE, EMAIL, CELL\_PHONE)  VALUES  (CUSTOMER\_ID\_SEQUENCE.NEXTVAL, 'Wong', 'Johnny', '62343213', '3018 Bedok North Street 5 03-34', 'Singapore', 'SG', 'johnnywong27@gmail.com', '97765433');  *-- ROW 2*  INSERT INTO ORARENT\_CUSTOMERS(CUSTOMER\_ID, LAST\_NAME, FIRST\_NAME, HOME\_PHONE, ADDRESS, CITY, STATE, EMAIL, CELL\_PHONE)  VALUES  (CUSTOMER\_ID\_SEQUENCE.NEXTVAL, 'West', 'Kanye', '65332266', '2021 Bt Batok St 23 #04-198', 'Singapore', 'SG', 'kanyewest2005@gmail.com', '95623014');  *-- ROW 3*  INSERT INTO ORARENT\_CUSTOMERS(CUSTOMER\_ID, LAST\_NAME, FIRST\_NAME, HOME\_PHONE, ADDRESS, CITY, STATE, EMAIL, CELL\_PHONE)  VALUES  (CUSTOMER\_ID\_SEQUENCE.NEXTVAL, 'Parker', 'Peter', '67355160', '143 Cecil Street 18-00 Gb Building', 'Singapore', 'SG', 'peterparker@yahoo.com', '88265973');  *-- ROW 4*  INSERT INTO ORARENT\_CUSTOMERS(CUSTOMER\_ID, LAST\_NAME, FIRST\_NAME, HOME\_PHONE, ADDRESS, CITY, STATE, EMAIL, CELL\_PHONE)  VALUES  (CUSTOMER\_ID\_SEQUENCE.NEXTVAL, 'Griffin', 'Peter', '62884568', 'BLK 119 ALJUNIED AVENUE 2, #01-K4', 'Singapore', 'SG', 'petergriffin69@gmail.com', '82215107');  *-- ROW 5*  INSERT INTO ORARENT\_CUSTOMERS(CUSTOMER\_ID, LAST\_NAME, FIRST\_NAME, HOME\_PHONE, ADDRESS, CITY, STATE, EMAIL, CELL\_PHONE)  VALUES  (CUSTOMER\_ID\_SEQUENCE.NEXTVAL, 'Tate', 'Andrew', '62743489', '101 Boon Keng Rd #06-01', 'Singapore', 'SG', 'andrewtate112@yahoo.com', '86942069');  *-- ROW 6*  INSERT INTO ORARENT\_CUSTOMERS(CUSTOMER\_ID, LAST\_NAME, FIRST\_NAME, HOME\_PHONE, ADDRESS, CITY, STATE, EMAIL, CELL\_PHONE)  VALUES  (CUSTOMER\_ID\_SEQUENCE.NEXTVAL, 'Marshall', 'Bruce', '69081243', '165 Marine Parade #03-01', 'Singapore', 'SG', 'brucemarshall22@gmail.com', '98865412');  SELECT \* FROM ORARENT\_CUSTOMERS; |

|  |
| --- |
| **ORARENT\_MEDIA** |
| *-- ORARENT MEDIA INSERTS*  *-- ROW 1*  INSERT INTO ORARENT\_MEDIA(MEDIA\_ID, FORMAT, TITLE\_ID)  VALUES (MEDIA\_ID\_SEQUENCE.NEXTVAL, 'DVD', 6);  *-- ROW 2*  INSERT INTO ORARENT\_MEDIA(MEDIA\_ID, FORMAT, TITLE\_ID)  VALUES (MEDIA\_ID\_SEQUENCE.NEXTVAL, 'VHS', 6);  *-- ROW 3*  INSERT INTO ORARENT\_MEDIA(MEDIA\_ID, FORMAT, TITLE\_ID)  VALUES (MEDIA\_ID\_SEQUENCE.NEXTVAL, 'DVD', 5);  *-- ROW 4*  INSERT INTO ORARENT\_MEDIA(MEDIA\_ID, FORMAT, TITLE\_ID)  VALUES (MEDIA\_ID\_SEQUENCE.NEXTVAL, 'VHS', 5);  *-- ROW 5*  INSERT INTO ORARENT\_MEDIA(MEDIA\_ID, FORMAT, TITLE\_ID)  VALUES (MEDIA\_ID\_SEQUENCE.NEXTVAL, 'DVD', 4);  *-- ROW 6*  INSERT INTO ORARENT\_MEDIA(MEDIA\_ID, FORMAT, TITLE\_ID)  VALUES (MEDIA\_ID\_SEQUENCE.NEXTVAL, 'VHS', 4);  SELECT \* FROM ORARENT\_MEDIA; |
| **ORARENT\_MOVIES** |
| *-- ORARENT MOVIES INSERTS*  *-- ROW 1*  INSERT INTO ORARENT\_MOVIES(TITLE\_ID,TITLE,DESCRIPTION,RATING,CATEGORY,RELEASE\_DATE)  VALUES (TITLE\_ID\_SEQUENCE.NEXTVAL, 'Forrest Gump', 'Forrest Gump, while not intelligent, has accidentally been present at many historic moments, but his true love, Jenny Curran, eludes him.', 'PG13', 'DRAMA', '6-JUN-1994');  *-- ROW 2*  INSERT INTO ORARENT\_MOVIES(TITLE\_ID,TITLE,DESCRIPTION,RATING,CATEGORY,RELEASE\_DATE)  VALUES (TITLE\_ID\_SEQUENCE.NEXTVAL, 'The Truman Show', 'An insurance salesman/adjuster discovers his entire life is actually a reality TV show.', 'PG', 'DRAMA', '5-JUN-1998');  *-- ROW 3*  INSERT INTO ORARENT\_MOVIES(TITLE\_ID,TITLE,DESCRIPTION,RATING,CATEGORY,RELEASE\_DATE)  VALUES (TITLE\_ID\_SEQUENCE.NEXTVAL, 'Mission Impossible', 'An American agent, under false suspicion of disloyalty, must discover and expose the real spy without the help of his organization.', 'PG13', 'ACTION', '22-JUN-1996');  *-- ROW 4*  INSERT INTO ORARENT\_MOVIES(TITLE\_ID,TITLE,DESCRIPTION,RATING,CATEGORY,RELEASE\_DATE)  VALUES (TITLE\_ID\_SEQUENCE.NEXTVAL, 'The Matrix', 'A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.', 'R', 'ACTION', '22-APR-1999');  *-- ROW 5*  INSERT INTO ORARENT\_MOVIES(TITLE\_ID,TITLE,DESCRIPTION,RATING,CATEGORY,RELEASE\_DATE)  VALUES (TITLE\_ID\_SEQUENCE.NEXTVAL, 'Shrek', 'After his swamp is filled with magical creatures, an ogre agrees to rescue a princess for a villainous lord in order to get his land back.', 'PG', 'COMEDY', '07-JUN-2001');  *-- ROW 6*  INSERT INTO ORARENT\_MOVIES(TITLE\_ID,TITLE,DESCRIPTION,RATING,CATEGORY,RELEASE\_DATE)  VALUES (TITLE\_ID\_SEQUENCE.NEXTVAL, 'Avatar', 'A paraplegic marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.', 'PG13', 'ACTION', '18-DEC-2009');  SELECT \* FROM ORARENT\_MOVIES;  **Text  Description automatically generated** |
| **ORARENT\_RENTAL\_HISTORY** |
| *-- ORARENT RENTAL HISTORY INSERTS*  *-- ROW 1*  INSERT INTO ORARENT\_RENTAL\_HISTORY(MEDIA\_ID, RENTAL\_DATE, CUSTOMER\_ID, RETURN\_DATE)  VALUES ( 92, '01-JAN-2021', 101, '05-JAN-2021');    *-- ROW 2*  INSERT INTO ORARENT\_RENTAL\_HISTORY(MEDIA\_ID, RENTAL\_DATE, CUSTOMER\_ID, RETURN\_DATE)  VALUES ( 93, '01-JAN-2021', 102, '05-JAN-2021');  *-- ROW 3*  INSERT INTO ORARENT\_RENTAL\_HISTORY(MEDIA\_ID, RENTAL\_DATE, CUSTOMER\_ID, RETURN\_DATE)  VALUES (94, '19-SEP-2022', 103, '22-SEP-2022');  *-- ROW 4*  INSERT INTO ORARENT\_RENTAL\_HISTORY(MEDIA\_ID, RENTAL\_DATE, CUSTOMER\_ID, RETURN\_DATE)  VALUES (95, '03-MAR-2022', 104, '07-MAR-2022');  *-- ROW 5*  INSERT INTO ORARENT\_RENTAL\_HISTORY(MEDIA\_ID, RENTAL\_DATE, CUSTOMER\_ID, RETURN\_DATE)  VALUES (92, '09-DEC-2022', 105, NULL);  *-- ROW 6*  INSERT INTO ORARENT\_RENTAL\_HISTORY(MEDIA\_ID, RENTAL\_DATE, CUSTOMER\_ID, RETURN\_DATE)  VALUES (95, '09-DEC-2022', 105, NULL);  SELECT \* FROM ORARENT\_RENTAL\_HISTORY;  A picture containing table  Description automatically generated |
| **ORARENT\_STAR\_BILLINGS** |
| *-- ORARENT STAR BILLINGS INSERTS*  *-- ROW 1*  INSERT INTO ORARENT\_STAR\_BILLINGS(ACTOR\_ID, TITLE\_ID, COMMENTS)  VALUES (1001, 1, 'Tom Hanks plays the role of Forrest Gump');  *-- ROW 2*  INSERT INTO ORARENT\_STAR\_BILLINGS(ACTOR\_ID, TITLE\_ID, COMMENTS)  VALUES (1002, 2, 'Jim Carrey is Truman Burbank');  *-- ROW 3*  INSERT INTO ORARENT\_STAR\_BILLINGS(ACTOR\_ID, TITLE\_ID, COMMENTS)  VALUES (1003, 4, 'Keanu Reeves is Neo in The Matrix');  *-- ROW 4*  INSERT INTO ORARENT\_STAR\_BILLINGS(ACTOR\_ID, TITLE\_ID, COMMENTS)  VALUES (1004, 6, 'Sam Worthington is Jake Sully in Avatar');  *-- ROW 5*  INSERT INTO ORARENT\_STAR\_BILLINGS(ACTOR\_ID, TITLE\_ID, COMMENTS)  VALUES (1005, 5, 'Eddie Murphy is Donkey in Shrek');  *-- ROW 6*  INSERT INTO ORARENT\_STAR\_BILLINGS(ACTOR\_ID, TITLE\_ID, COMMENTS)  VALUES (1006, 3, 'Tom Cruise is Ethan Hunt');  SELECT \* FROM ORARENT\_STAR\_BILLINGS;  Graphical user interface, application  Description automatically generated |

1. Create an index on the last\_name column of the Customers table



*-- creating indexes on the last\_name column of the Customers table.*

CREATE INDEX **CUSTOMERS\_LAST\_NAME\_IDX**

ON ORARENT\_CUSTOMERS(LAST\_NAME);

*-- verify that the index was created*

SELECT INDEX\_NAME, TABLE\_NAME

FROM USER\_INDEXES

where TABLE\_NAME = 'ORARENT\_CUSTOMERS';

1. Create a view called TITLE\_UNAVAIL to show the movie titles and media\_id of the media not returned yet.

*-- Create a view called TITLE\_UNAVAIL to show the movie titles and media\_id of the media not returned yet. The view should not allow any DML operations.*

CREATE VIEW **TITLE\_UNAVAIL** AS

SELECT TITLE, MEDIA\_ID

FROM ORARENT\_MEDIA NATURAL JOIN ORARENT\_MOVIES

WHERE MEDIA\_ID IN (SELECT MEDIA\_ID FROM ORARENT\_RENTAL\_HISTORY WHERE RETURN\_DATE IS NULL)

WITH READ ONLY;

*-- SQL STATEMENT TO VERIFY THE CREATION OF VIEW ON THE CUSTOMERS TABLE*

SELECT VIEW\_NAME , TEXT , READ\_ONLY

FROM USER\_VIEWS

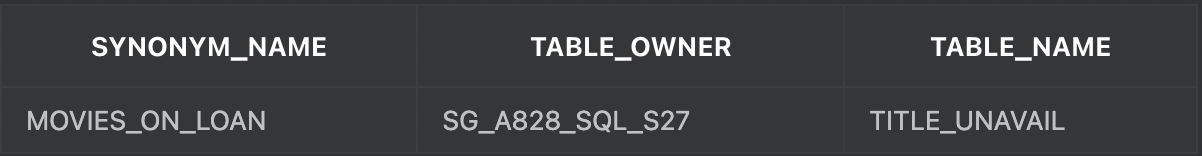
WHERE VIEW\_NAME = 'TITLE\_UNAVAIL';

Graphical user interface, text

Description automatically generated

*Note: Note: The movies that will be returned by a select statement on the view are "Shrek" and "Avatar", as their return dates are null (indicating that they are currently on loan).*

1. Create a synonym for the TITLE\_UNAVAIL view.



*-- SQL STATEMENT TO VERIFY THE CREATION OF SYNONYM ON THE CUSTOMERS TABLE*

SELECT SYNONYM\_NAME, TABLE\_OWNER, TABLE\_NAME

FROM USER\_SYNONYMS

WHERE SYNONYM\_NAME = 'MOVIES\_ON\_LOAN';

*-- Create a synonym for the TITLE\_UNAVAIL view.*

SELECT SYNONYM\_NAME, TABLE\_OWNER, TABLE\_NAME

FROM USER\_SYNONYMS

WHERE SYNONYM\_NAME = 'MOVIES\_ON\_LOAN';

# Assumptions

1. We have enabled caching for the **CUSTOMER\_ID\_SEQUENCE** to improve performance in the busy orarental video store. With frequent customer visits and many transactions occurring, this will help ensure efficient ID generation.
2. To ensure unique primary keys, we have enabled the *NOCYCLE* option on our sequences. This prevents repetition and eliminates the need for a *MINVALUE*, as the sequence will not recycle once it reaches its maximum value.
3. We have set the *MAXVALUE* to its default (10^27) as we do not have a specific estimate for the number of customers who will visit the store. This will ensure that our sequence has a sufficiently high maximum value to generate unique primary keys for an uncertain volume of customers.