

OTT Management System

Table Description

1) Subscription Table

create table subscription (subscription_id int primary key, subscription_type varchar(20), amount int) ;

desc subscription ;

TABLE SUBSCRIPTION

Column	Null?	Type
SUBSCRIPTION_ID	NOT NULL	NUMBER
SUBSCRIPTION_TYPE	-	VARCHAR2(20)
AMOUNT	-	NUMBER

2) Has_payment Table

alter table has_payment add

(
 payment_id references payment_type(payment_id)
);

alter table has_payment add

(
 subscription_id references subscription(subscription_id)
);

alter table has_payment add primary key (subscription_id, payment_id);

desc has_payment;

TABLE HAS_PAYMENT

Column	Null?	Type
PAYMENT_ID	NOT NULL	NUMBER
SUBSCRIPTION_ID	NOT NULL	NUMBER

3) Userott Table

```
create table userOtt (user_id int primary key, name varchar(50),age int,phone varchar(20),city  
varchar(20),state varchar(20), street varchar(20), subscription_start_date  
date,subscription_end_date date);
```

```
alter table userott add  
(  
    subscription_id references subscription(subscription_id)  
);
```

```
desc userOtt;
```

TABLE USEROTT

Column	Null?	Type
USER_ID	NOT NULL	NUMBER
NAME	-	VARCHAR2(50)
AGE	-	NUMBER
PHONE	-	VARCHAR2(20)
CITY	-	VARCHAR2(20)
STATE	-	VARCHAR2(20)
STREET	-	VARCHAR2(20)
SUBSCRIPTION_START_DATE	-	DATE
SUBSCRIPTION_END_DATE	-	DATE
SUBSCRIPTION_ID	-	NUMBER

4) Watches Table

```
create table watches (profile_id int, is_completed char(1), no_of_views int ,  
remaining_watch_time timestamp);
```

```
alter table watches add  
(  
    user_id references userott(user_id)
```

);

alter table watches add

(
content_id references content_type(content_id)
);

alter table has_payment add primary key (user_id, content_id, profile_id);

desc watches;

TABLE WATCHES

Column	Null?	Type
IS_COMPLETED	-	CHAR(1)
NO_OF_VIEWS	-	NUMBER
REMAINING_WATCH_TIME	-	TIMESTAMP(6)
USER_ID	NOT NULL	NUMBER
CONTENT_ID	NOT NULL	NUMBER
PROFILE_ID	NOT NULL	NUMBER

5) Device Enrolled Table

create table device_enrolled (device_id int, device_name varchar(20));

alter table device_enrolled add

(
user_id references userott(user_id)
);

alter table has_payment add primary key (user_id, device_id);

desc device_enrolled;

TABLE DEVICE_ENROLLED

Column	Null?	Type
DEVICE_ID	NOT NULL	NUMBER
DEVICE_NAME	-	VARCHAR2(20)
USER_ID	NOT NULL	NUMBER

6) Downloads Table

create table downloads (profile_id int, download_id int);

alter table downloads add

(
 user_id references userott(user_id)
);

alter table downloads add primary key (user_id, download_id, profile_id);

desc downloads;

TABLE DOWNLOADS

Column	Null?	Type
DOWNLOAD_ID	NOT NULL	NUMBER
USER_ID	NOT NULL	NUMBER
PROFILE_ID	NOT NULL	NUMBER

7) Watchlist Table

create table watchlist (profile_id int ,watchlist_id int);

alter table watchlist add

(
 user_id references userott(user_id)
);

alter table watchlist add primary key (user_id, watchlist_id, profile_id);

desc watchlist;

TABLE WATCHLIST

Column	Null?	Type
WATCHLIST_ID	NOT NULL	NUMBER
USER_ID	NOT NULL	NUMBER
PROFILE_ID	NOT NULL	NUMBER

8) Profile Table

create table profile(profile_id int,profile_name varchar(20));

alter table profile add

(
user_id references userott(user_id)
);

alter table profile add primary key (user_id, profile_id);

desc profile;

TABLE PROFILE

Column	Null?	Type
PROFILE_ID	NOT NULL	NUMBER
PROFILE_NAME	-	VARCHAR2(20)
USER_ID	NOT NULL	NUMBER

9) Cast Table

create table cast (cast_name varchar(20) primary key);

alter table cast add

(
content_id references content_type(content_id)
);

```
alter table cast add primary key (cast_name, content_id);
```

```
desc cast;
```

TABLE CAST

Column	Null?	Type
CAST_NAME	NOT NULL	VARCHAR2(20)
CONTENT_ID	NOT NULL	NUMBER

10) Content_Type Table

```
create table content_type (content_id int primary key,name varchar(20),genre  
varchar(20),user_rating int, imdb_rating int, duration timestamp,season_no int,episode_no  
int,t_type varchar(20));
```

```
alter table content_type add
```

```
(  
    subscription_id references subscription(subscription_id)  
);
```

```
desc content_type;
```

TABLE CONTENT_TYPE

Column	Null?	Type
CONTENT_ID	NOT NULL	NUMBER
NAME	-	VARCHAR2(20)
GENRE	-	VARCHAR2(20)
USER_RATING	-	NUMBER
IMDB_RATING	-	NUMBER
DURATION	-	TIMESTAMP(6)
SEASON_NO	-	NUMBER
EPISODE_NO	-	NUMBER
T_TYPE	-	VARCHAR2(20)
SUBSCRIPTION_ID	-	NUMBER

11) Payment_Type Table

```
create table payment_type (payment_id int primary key,payment_date timestamp,amount_due  
int,p_type varchar(20), creditcard_no varchar(20), debitcard_no varchar(20),netbanking_id  
varchar(20),upi_id varchar(20));
```

```
desc payment_type;
```

TABLE PAYMENT_TYPE

Column	Null?	Type
PAYMENT_ID	NOT NULL	NUMBER
PAYMENT_DATE	-	TIMESTAMP(6)
AMOUNT_DUE	-	NUMBER
P_TYPE	-	VARCHAR2(20)
CREDITCARD_NO	-	VARCHAR2(20)
DEBITCARD_NO	-	VARCHAR2(20)
NETBANKING_ID	-	VARCHAR2(20)
UPI_ID	-	VARCHAR2(20)
USER_ID	-	NUMBER

DDL(Data Definition Language)

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema.

CREATE

```
create table subscription (subscription_id int primary key, subscription_type varchar(20),amount  
int) ;
```

TABLE SUBSCRIPTION

Column	Null?	Type
SUBSCRIPTION_ID	NOT NULL	NUMBER
SUBSCRIPTION_TYPE	-	VARCHAR2(20)
AMOUNT	-	NUMBER

DROP

drop table subscription

```
Table dropped.
```

ALTER

alter table subscription drop column subscription_type;

TABLE SUBSCRIPTION

Column	Null?	Type
SUBSCRIPTION_ID	NOT NULL	NUMBER
AMOUNT	-	NUMBER

TRUNCATE

truncate table user;

```
Table truncated.
```


COMMENT

```
-- single line comment
-- another comment
SELECT * FROM Subscription;
```

```
/* multi line comment
another comment */
```

```
SELECT * FROM Subscription;
```

RENAME

ALTER TABLE Subscription

RENAME COLUMN subscription_type TO subscription_types;

TABLE SUBSCRIPTION

Column	Null?	Type
SUBSCRIPTION_ID	NOT NULL	NUMBER
AMOUNT	-	NUMBER
SUBSCRIPTION_TYPES	-	VARCHAR2(20)

DML(Data Manipulation Language)

The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

INSERT

```
INSERT INTO subscription
(subscription_id,amount,subscription_types)
VALUES
(1,0,'Free');
```

```
1 row(s) inserted.
```

SELECT

```
select * from subscription
```

SUBSCRIPTION_ID	AMOUNT	SUBSCRIPTION_TYPES
1	0	Free

UPDATE

Update subscription set subscription_types='unpaid'
where subscription_id=1;

```
Select * from subscription;
```

SUBSCRIPTION_ID	AMOUNT	SUBSCRIPTION_TYPES
1	0	unpaid

DELETE

```
delete from subscription where subscription_id=1;
```

```
1 row(s) deleted.
```

DCL(Data Control Language)

DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.

GRANT:

Used to provide any user access privileges or other privileges for the database.

Example:

```
GRANT CREATE TABLE TO 'Ishani';
```

REVOKE:

Used to take back permissions from any user.

```
REVOKE CREATE TABLE FROM 'Ishani';
```

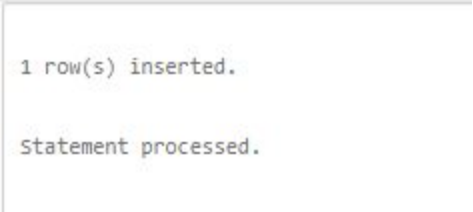
TCL(transaction Control Language) :

TCL commands deal with the transaction within the database.

COMMIT

Commits a Transaction.

```
INSERT INTO subscription  
(subscription_id,amount,subscription_types)  
VALUES  
(2,0,'Premium');  
COMMIT ;
```



```
1 row(s) inserted.  
  
statement processed.
```

```
delete from subscription where subscription_id=2;  
commit ;  
select * from subscription ;
```

1 row(s) deleted.

Statement processed.

SUBSCRIPTION_ID	AMOUNT	SUBSCRIPTION_TYPES
1	0	unpaid

[Download CSV](#)

ROLLBACK

Rollbacks a transaction in case of any error occurs.

```
delete from subscription where subscription_id=2;
rollback ;
select * from subscription ;
```

1 row(s) deleted.

Statement processed.

SUBSCRIPTION_ID	AMOUNT	SUBSCRIPTION_TYPES
1	0	unpaid
2	0	Premium

[Download CSV](#)

2 rows selected.

SAVEPOINT

Sets a savepoint within a transaction.

//Insert statements

```
Insert into userott(user_id , name ,age,phone,city ,state,
street,subscription_start_date,subscription_end_date, subscription_id ) values(1, 'Ishani Vaghela',
23,'9843542314','Chennai','Tamil Nadu','2',DATE '2021-12-17',DATE '2022-12-17',1);
```

```
Insert into userott(user_id , name ,age,phone,city ,state,
street,subscription_start_date,subscription_end_date, subscription_id ) values(2, 'Ritha',
23,'9843542313','Chennai','Tamil Nadu','2',DATE '2021-12-17',DATE '2022-12-17',1);
```

```
select * from userott
```

USER_ID	NAME	AGE	PHONE	CITY	STATE	STREET	SUBSCRIPTION_START_DATE	SUBSCRIPTION_END_DATE	SUBSCRIPTION_ID
2	Ritha	23	9843542313	Chennai	Tamil Nadu	2	17-DEC-21	17-DEC-22	1
1	Ishani Vaghela	23	9843542314	Chennai	Tamil Nadu	2	17-DEC-21	17-DEC-22	1

[Download CSV](#)

2 rows selected.

```
Delete from userott where user_id=2;
```

```
Savepoint s1;
```

```
Delete from userott where user_id=1;
```

```
Savepoint s2;
```

```
Rollback to s1;
```

```
Select * from userott;
```

USER_ID	NAME	AGE	PHONE	CITY	STATE	STREET	SUBSCRIPTION_START_DATE	SUBSCRIPTION_END_DATE	SUBSCRIPTION_ID
1	Ishani Vaghela	23	9843542314	Chennai	Tamil Nadu	2	17-DEC-21	17-DEC-22	1

[Download CSV](#)

VIEW

In SQL, a view is a virtual table based on the result-set of an SQL statement.

```
CREATE VIEW user_view AS
```

```
SELECT user_id, name, age,phone
```

```
FROM userott;
```

```
select * from user_view;
```

View created.

USER_ID	NAME	AGE	PHONE
1	Ishani Vaghela	23	9843542314
2	Ritha	23	9843542313

[Download CSV](#)

2 rows selected.

AGGREGATE FUNCTIONS

The aggregate functions are COUNT, MAX, MIN, SUM, AVG. They are used to group the result-set by one or more columns.

```
select * from userott;
```

USER_ID	NAME	AGE	PHONE	CITY	STATE	STREET	SUBSCRIPTION_START_DATE	SUBSCRIPTION_END_DATE	SUBSCRIPTION_ID
3	Anisha	15	9844342314	Mumbai	Maharashtra	2	17-DEC-21	17-DEC-22	1
4	Harshini	28	9888842313	Kolkata	West Bengal	2	17-DEC-21	17-DEC-22	1
5	Anu	12	9843599314	Chennai	Tamil Nadu	2	17-DEC-21	17-DEC-22	1
6	Abi	32	984354813	Chennai	Tamil Nadu	2	17-DEC-21	17-DEC-22	1
1	Ishani Vaghela	23	9843542314	Chennai	Tamil Nadu	2	17-DEC-21	17-DEC-22	1
2	Ritha	23	9843542313	Chennai	Tamil Nadu	2	17-DEC-21	17-DEC-22	1

[Download CSV](#)

6 rows selected.

i) Count

```
select age,count(user_id) as Age_wise_count from userott group by age order by age ;
```

AGE	AGE_WISE_COUNT
12	1
15	1
23	2
28	1
32	1

[Download CSV](#)

5 rows selected.

ii) Max & Min

```
select max(age) as max_age,min(age) as min_age from userott ;
```

[Download CSV](#)

```
select avg(age) as average_age from userott ;
```

[Download CSV](#)

```
select * from userott u1 ,subscription s1 where u1.subscription_id=s1.subscription_id and
s1.subscription_id in (select subscription_id from subscription where
subscription_types='unpaid');
```