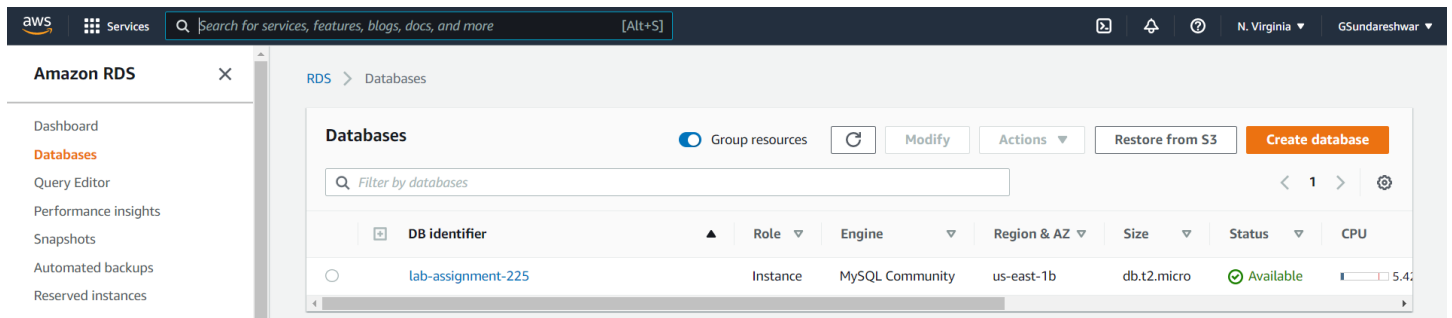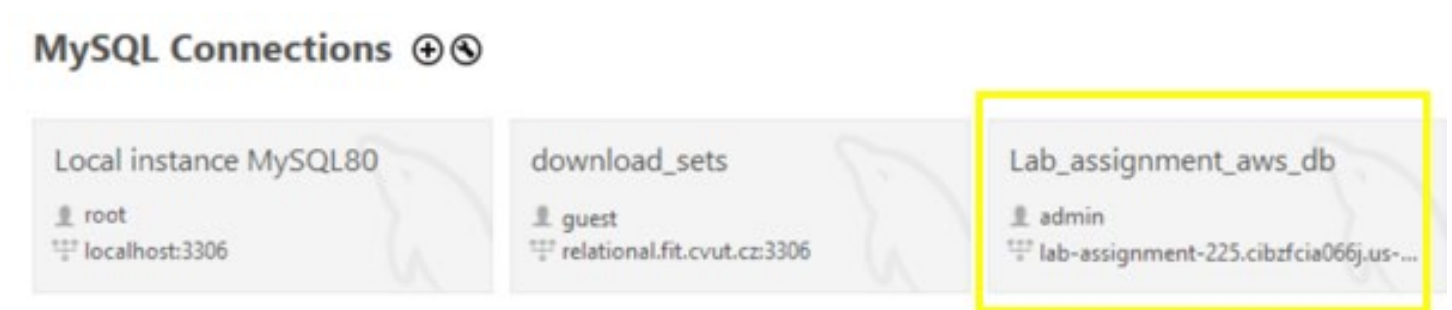# AWS DB,TABLE CREATION, ER – QUERIES AND SCREENSHOTS

AWS DATABASE DETAILS
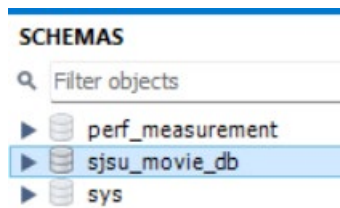

CONFIGURING AWS DB IN MYSQL

DATABASE CREATED IN MYSQL

TABLE CREATION QUERIES

ACTOR DETAILS:

Key	Column	Datatype	Query
FK	Rank_ID	int	CREATE TABLE Actor_Details
	Actor_Name	varchar(100)	(Rank_ID int, Actor_Name varchar(100), FOREIGN KEY (Rank_ID) references sjsu_movie_db.Movie_data(Rank_ID))

GENRE DETAILS:

Key	Column	Datatype	Query
FK	Rank_ID	int	CREATE TABLE Genre_Details
	Genre	varchar(100)	(Rank_ID int, Genre varchar(100), FOREIGN KEY (Rank_ID) references sjsu_movie_db.Movie_data(Rank_ID))

MOVIE DATA:

Key	Column	Datatype	Query
PK	Rank_ID	int AI	CREATE TABLE Movie_data
	Title	varchar(100)	(Rank_ID int PRIMARY KEY AUTO_INCREMENT not null,
	Description	varchar(300)	Title varchar(100),
	Director	varchar(20)	Description varchar(300),
	Year	int	Director varchar(20),
	Runtime	int	Year int,
	Rating	float	Runtime int,
	Votes	bigint	Rating float,
	Revenue	float	Votes bigint,
	Metascore	int	Revenue float, Metascore int)

CUSTOMER DETAILS:

Key	Column	Datatype	Query
PK	customer_id	varchar(15)	CREATE TABLE customer_details
	cust_ssn	varchar(12)	(customer_id varchar(15) PRIMARY KEY not null,
	cust_first_name	varchar(40)	cust_ssn varchar(12),
	cust_middle_name	varchar(40)	cust_first_name varchar(40),
	cust_last_name	varchar(40)	cust_middle_name varchar(40),
	cust_phone_no	varchar(24)	cust_last_name varchar(40),
	cust_email	varchar(60)	cust_phone_no varchar(24),
	cust_add_line_1	varchar(100)	cust_email varchar(60),
	cust_city	varchar(50)	cust_add_line_1 varchar(100),
	cust_state	varchar(50)	cust_city varchar(50),
	cust_country	varchar(50)	cust_state varchar(50),
	cust_zipcode	int	cust_country varchar(50), cust_zipcode int);

USER DETAILS:

Key	Column	Datatype	Query
PK	user_id	varchar(15)	create table user_details
FK	customer_id	varchar(15)	(
	screen_no	int	user_id varchar(15) PRIMARY KEY NOT NULL, customer_id varchar(15), screen_no int, FOREIGN KEY (customer_id) references sjsu_movie_db.customer_details(customer_id));

CUSTOMER COMPLAINTS:

Key	Column	Datatype	Query
PK	complaint_id	varchar(10)	CREATE TABLE cust_complaints
	complaint_creation_date	date	(
FK	customer_id	varchar(15)	complaint_id varchar(10) NOT NULL PRIMARY KEY,
FK	emp_id	varchar(10)	complaint_creation_date date, customer_id varchar(15),
	severity	varchar(40)	emp_id varchar(10),
	complaint_description	varchar(200)	severity varchar(40),
	complaint_category	varchar(50)	complaint_description varchar(200),
	resolution_status	varchar(20)	complaint_category varchar(50),
	estimated_resolution_date	date	resolution_status varchar(20),
	close_date	date	estimated_resolution_date date, close_date date, FOREIGN KEY (emp_id) references sjsu_movie_db.employees(emp_id)), FOREIGN KEY (customer_id) references sjsu_movie_db.customer_details(customer_id));

EMPLOYEES:

Key	Column	Datatype	Query
PK	emp_id	varchar(15) PK	CREATE TABLE employees
	emp_ssn	varchar(12)	(
	emp_first_name	varchar(40)	emp_id varchar(15) PRIMARY KEY NOT NULL, emp_ssn varchar(12),
	emp_middle_name	varchar(40)	emp_first_name varchar(40),
	emp_last_name	varchar(40)	emp_middle_name varchar(40),
	emp_phone_number	varchar(24)	emp_last_name varchar(40), emp_phone_number varchar(24),
	emp_email	varchar(60)	emp_email varchar(60),
	employment_status	varchar(40)	employment_status varchar(40),
	emp_department	varchar(30)	emp_department varchar(30),
	emp_salary	int	emp_salary int, emp_position varchar(20)
	emp_position	varchar(20)	)

USER_WATCH_HISTORY:

Key	Column	Datatype	Query
FK	**user_id**	varchar(15)	CREATE TABLE user_watch_history
FK	**rank_id**	int	(
	watch_date	date	user_id varchar(15), rank_id int, watch_date date, FOREIGN KEY (user_id) references sjsu_movie_db.user_details(user_id), FOREIGN KEY (rank_id) references sjsu_movie_db.Movie_data(Rank_ID));

INVOICE DETAILS:

Key	Column	Datatype	Query
PK	**invoice_id**	varchar(10)	CREATE TABLE invoice_details
FK	customer_id	varchar(15)	(
	payment_method	varchar(30)	invoice_id varchar(10) NOT NULL PRIMARY KEY,
	total_amount	decimal(4,2)	customer_id varchar(15), payment_method varchar(30),
	payment_date	date	total_amount decimal(4,2) not null, payment_date date, FOREIGN KEY (customer_id) references sjsu_movie_db.customer_details(customer_id));

SCHEMA

SCHEMAS

🔍 Filter objects

▼ 🗄 sjsu_movie_db
  ▼ 🗇 Tables
    ▶ ▦ Actor_Details
    ▶ ▦ audit_Actor_Details
    ▶ ▦ audit_cust_complaints
    ▶ ▦ audit_customer_details
    ▶ ▦ audit_employees
    ▶ ▦ audit_Genre_Details
    ▶ ▦ audit_invoice_details
    ▶ ▦ audit_Movie_data
    ▶ ▦ audit_user_details
    ▶ ▦ audit_user_watch_history
    ▶ ▦ cust_complaints
    ▶ ▦ customer_details
    ▶ ▦ employees
    ▶ ▦ Genre_Details
    ▶ ▦ invoice_details
    ▶ ▦ Movie_data
    ▶ ▦ user_details
    ▶ ▦ user_watch_history
  🗇 Views
  ▶ 🗇 Stored Procedures
  🗇 Functions

SAMPLE TABLE CREATION SCREENSHOTS

MOVIE DATA:

```
1 •  create table Movie_data (
2      Rank_ID int primary key AUTO_INCREMENT not null,
3      Title varchar(100),
4      Description varchar(300),
5      Director varchar(20),
6      Year int,
7      Runtime int,
8      Rating float,
9      Votes bigint,
10     Revenue float,
11     Metascore int);
```

Output

#	Time	Action	Message	Duration / Fetch
✓	1 21:53:54	create table Movie_data (Rank_ID int primary key AUTO_INCREMENT not null, Title varchar(100), Des...	0 row(s) affected	0.110 sec

CUSTOMER DETAILS:

```
1 •  create table customer_details(
2      customer_id varchar(15) primary key not null,
3      cust_ssn varchar(12),
4      cust_first_name varchar(40),
5      cust_middle_name varchar(40),
6      cust_last_name varchar(40),
7      cust_phone_no varchar(24),
8      cust_email varchar(60),
9      cust_add_line_1 varchar(100),
10     cust_city varchar(50),
11     cust_state varchar(50),
12     cust_country varchar(50),
13     cust_zipcode int);
```

Output

#	Time	Action	Message	Duration / Fetch
✓	1 21:56:25	create table customer_details(customer_id varchar(15) primary key not null, cust_ssn varchar(12), cust_fi...	0 row(s) affected	0.141 sec

ACTOR DETAILS:

```
1   Create table Actor_Details (
2      Rank_ID int,
3      Actor_Name varchar(100),
4      foreign key (Rank_ID) references sjsu_movie_db.Movie_data(Rank_id));
```

Output

#	Time	Action	Message	Duration / Fetch
✓	1 17:03:16	Create table Actor_Details (Rank_ID int, Actor_Name varchar(100), foreign key (Rank_ID) references sjs...	0 row(s) affected	0.141 sec

ER DIAGRAM

cust_complaints
- complaint_id VARCHAR(10)
- complaint_creation_date DATE
- customer_id VARCHAR(15)
- emp_id VARCHAR(10)
- severity VARCHAR(40)
- complaint_description VARCHAR(200)
- complaint_category VARCHAR(50)
- resolution_status VARCHAR(20)
- estimated_resolution_date DATE
- close_date DATE

Indexes
- PRIMARY
- emp_id
- customer_id

employees
- emp_id VARCHAR(15)
- emp_ssn VARCHAR(12)
- emp_first_name VARCHAR(40)
- emp_middle_name VARCHAR(40)
- emp_last_name VARCHAR(40)
- emp_phone_number VARCHAR(24)
- emp_email VARCHAR(60)
- employment_status VARCHAR(40)
- emp_department VARCHAR(30)
- emp_salary INT
- emp_position VARCHAR(20)

Indexes
- PRIMARY

Genre_Details
- Rank_ID INT
- Genre VARCHAR(100)

Indexes
- Rank_ID

Movie_data
- Rank_ID INT
- Title VARCHAR(100)
- Description VARCHAR(300)
- Director VARCHAR(20)
- Year INT
- Runtime INT
- Rating FLOAT
- Votes BIGINT
- Revenue FLOAT
- Metascore INT

Indexes
- PRIMARY

audit_customer_details
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- customer_id VARCHAR(15)
- cust_ssn VARCHAR(12)
- cust_first_name VARCHAR(40)
- cust_middle_name VARCHAR(40)
- cust_last_name VARCHAR(40)
- cust_phone_no VARCHAR(24)
- cust_email VARCHAR(60)
- cust_add_line_1 VARCHAR(100)
- cust_city VARCHAR(50)
- cust_state VARCHAR(50)
- cust_country VARCHAR(50)
- cust_zipcode INT

Indexes
- PRIMARY
- auditTimestamp

audit_Movie_data
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- Rank_ID INT
- Title VARCHAR(100)
- Description VARCHAR(300)
- Director VARCHAR(20)
- Year INT
- Runtime INT
- Rating FLOAT
- Votes BIGINT
- Revenue FLOAT
- Metascore INT

Indexes
- PRIMARY
- auditTimestamp

audit_user_details
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- user_id VARCHAR(15)
- customer_id VARCHAR(15)
- screen_no INT

Indexes
- PRIMARY
- auditTimestamp

audit_Actor_Details
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- Rank_ID INT
- Actor_Name VARCHAR(100)

Indexes
- PRIMARY
- auditTimestamp

customer_details
- customer_id VARCHAR(15)
- cust_ssn VARCHAR(12)
- cust_first_name VARCHAR(40)
- cust_middle_name VARCHAR(40)
- cust_last_name VARCHAR(40)
- cust_phone_no VARCHAR(24)
- cust_email VARCHAR(60)
- cust_add_line_1 VARCHAR(100)
- cust_city VARCHAR(50)
- cust_state VARCHAR(50)
- cust_country VARCHAR(50)
- cust_zipcode INT

Indexes
- PRIMARY

user_details
- user_id VARCHAR(15)
- customer_id VARCHAR(15)
- screen_no INT

Indexes
- PRIMARY
- customer_id

Actor_Details
- Rank_ID INT
- Actor_Name VARCHAR(100)

Indexes
- Rank_ID

audit_cust_complaints
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- complaint_id VARCHAR(10)
- complaint_creation_date DATE
- customer_id VARCHAR(15)
- emp_id VARCHAR(10)
- severity VARCHAR(40)
- complaint_description VARCHAR(200)
- complaint_category VARCHAR(50)
- resolution_status VARCHAR(20)
- estimated_resolution_date DATE
- close_date DATE

Indexes
- PRIMARY
- auditTimestamp

audit_employees
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- emp_id VARCHAR(15)
- emp_ssn VARCHAR(12)
- emp_first_name VARCHAR(40)
- emp_middle_name VARCHAR(40)
- emp_last_name VARCHAR(40)
- emp_phone_number VARCHAR(24)
- emp_email VARCHAR(60)
- employment_status VARCHAR(40)
- emp_department VARCHAR(30)
- emp_salary INT
- emp_position VARCHAR(20)

Indexes
- PRIMARY
- auditTimestamp

audit_user_watch_history
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- user_id VARCHAR(15)
- rank_id INT
- watch_date DATE

Indexes
- PRIMARY
- auditTimestamp

user_watch_history
- user_id VARCHAR(15)
- rank_id INT
- watch_date DATE

Indexes
- user_watch_history_ibfk_1
- user_watch_history_ibfk_2

audit_Genre_Details
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- Rank_ID INT
- Genre VARCHAR(100)

Indexes
- PRIMARY
- auditTimestamp

audit_invoice_details
- auditAction ENUM(...)
- auditTimestamp DATETIME
- auditId INT
- Performed By VARCHAR(200)
- invoice_id VARCHAR(10)
- customer_id VARCHAR(15)
- payment_method VARCHAR(30)
- total_amount DECIMAL(4,2)
- payment_date DATE

Indexes
- PRIMARY
- auditTimestamp

invoice_details
- invoice_id VARCHAR(10)
- customer_id VARCHAR(15)
- payment_method VARCHAR(30)
- total_amount DECIMAL(4,2)
- payment_date DATE

Indexes
- PRIMARY
- invoice_details_ibfk_1

Relationships:
- handles
- belongs to
- can raise
- linked to
- acted in
- have
- contains
- pays by