**ADVANCE DATABASE MANAGEMENT SYSTEM PROJECT REPORT**

**Rajashekar Reddy Patlori, Lokesh Anjaneya Pothana, Rajashekhar Reddy Chinthalapalli, Kandapu Ajita, and Yogeshwar Reddy Kandati**

**Project Aim**

This project aims to add Oscar awards data and develop a system that enhances the existing movie database (SQL DB).

**Project Requirement**

The focus of the project is to create two Oscar award tables one is for actors and the other is for writers & directors to specify the artists who are nominees and winners of Oscar Awards for the movies which are in the movie table.

**Database Design**

Partitioned data as actors, writers, and directors referring to the data set gathered from the Kaggle website. Developed two tables in SQL111 that are the “Actors Oscar Award” and “Job Oscar Award”. The actors’ Oscar Award table specifies the data of artists who are nominees and winners of the Oscar Awards and their role names. The Job Oscar Award table provides information on writers and directors who are the nominees and winners of the Oscar Awards.

In the process of building the Oscar awards tables sorted all the movies, and person names from the data set which are unique with the respective tables of movies, movie\_actors, and movie\_jobs. And created two different excel sheets for actors and jobs that are nominated and won awards and then exported to the database.

These two tables contain the details of the movie name, release year, ceremony year, person name, category, Role Name, Job Code, and award status. We have built the tables with reference to the tables like movies, movie\_actors, movies jobs, and persons using columns like movie\_guid, person\_name, movie\_title, and person\_guid.

**Screenshots**

ACTOR\_OSCAR\_AWARDS table details

Table

Description automatically generated

ACTOR\_OSCAR\_AWARDS model

Text

Description automatically generated with medium confidence

ACTOR\_OSCAR\_AWARDS table constraints

Graphical user interface, application

Description automatically generated

ACTOR\_OSCAR\_AWARDS Data Sample

A picture containing graphical user interface

Description automatically generated

JOB\_OSCAR\_AWARDS table details

Graphical user interface, table

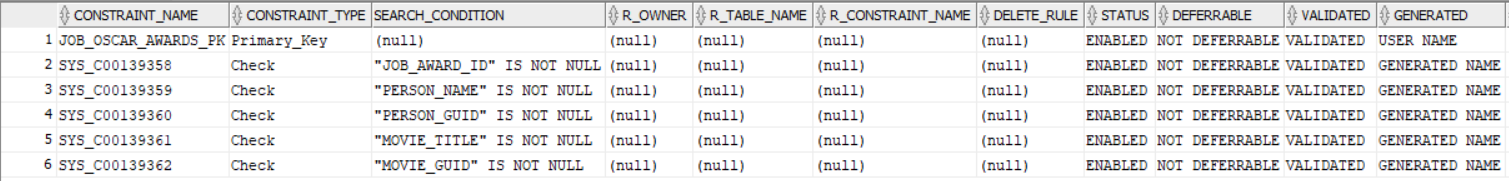
Description automatically generated with medium confidence

JOB\_OSCAR\_AWARDS model

A picture containing table

Description automatically generated

JOB\_OSCAR\_AWARDS table constraints



JOB\_OSCAR\_AWARDS Data Sample

A picture containing text

Description automatically generated

**Entity Relationship Diagram (Excluding all other tables)**

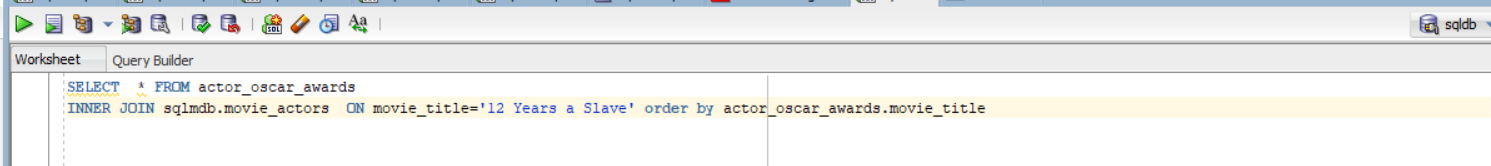
**Chart, box and whisker chart

Description automatically generated**

**Query writing**

**Query 1**

Display the list of actors who have been nominated and won Oscar awards for the movie “12 Years a Slave” movie.

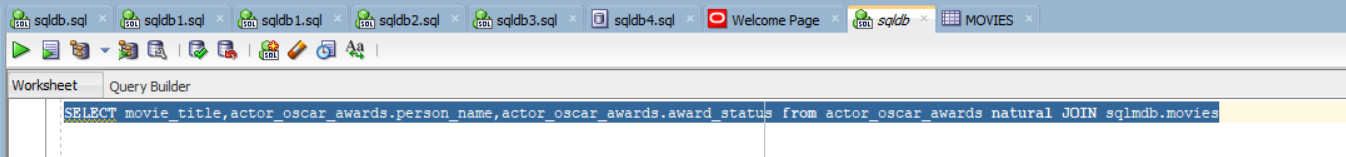
****

**A screenshot of a computer

Description automatically generated with medium confidence**

**Query 2**

Displays the list of all the actors their award status and the movie they have acted in.

****

**Graphical user interface, application

Description automatically generated**

**Query 3**

Displays the list of all the actors who have won the Oscar wards

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

**Query 4**

Displays the list of all the actors, writers, and directors who have been Nominated and won the Oscar wards.

Timeline

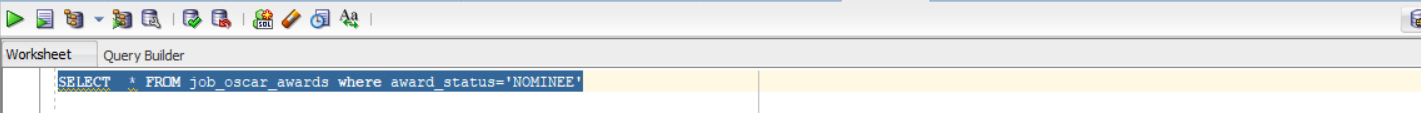
Description automatically generated with medium confidence

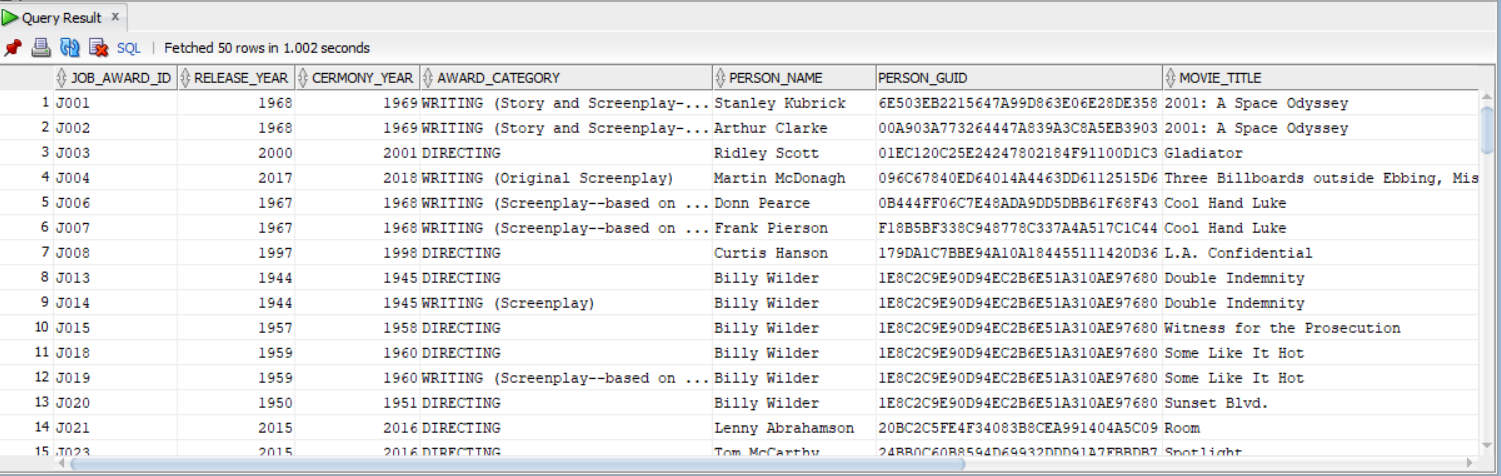
Graphical user interface, table

Description automatically generated

**Query 5**

Displays the list of all the writers, and directors who have been Nominated for the Oscar awards.





**Performance Tuning**

SQL tuning is the process of boosting up SQL queries to accelerate your server’s performance. The general purpose is to bring down the time it takes a user to receive a result after issuing a query and reduce the number of resources used to process a query.

**Index**

The index is a table that gives faster query results and data retrieval from the database. Index tables use indexing, an approach that makes data structures speed up a database query's searching time. Indexing makes database performance better and consumes lesser memory in the main memory.

Before indexing, we can notice that the scan is full-time, and costs are high

Graphical user interface, application

Description automatically generated

After creating the index and implementing the same select statement the cost and time has been reduced.

Graphical user interface, application

Description automatically generated

**Optimizer modes**

ALL\_ROWS is the default optimizer Getting all rows faster is good for untuned, high-volume batch systems.

FIRST\_ROWS Gets the first row faster this is good for untuned systems that process lots of single transactions.

Table

Description automatically generated

**Parallel Execution**

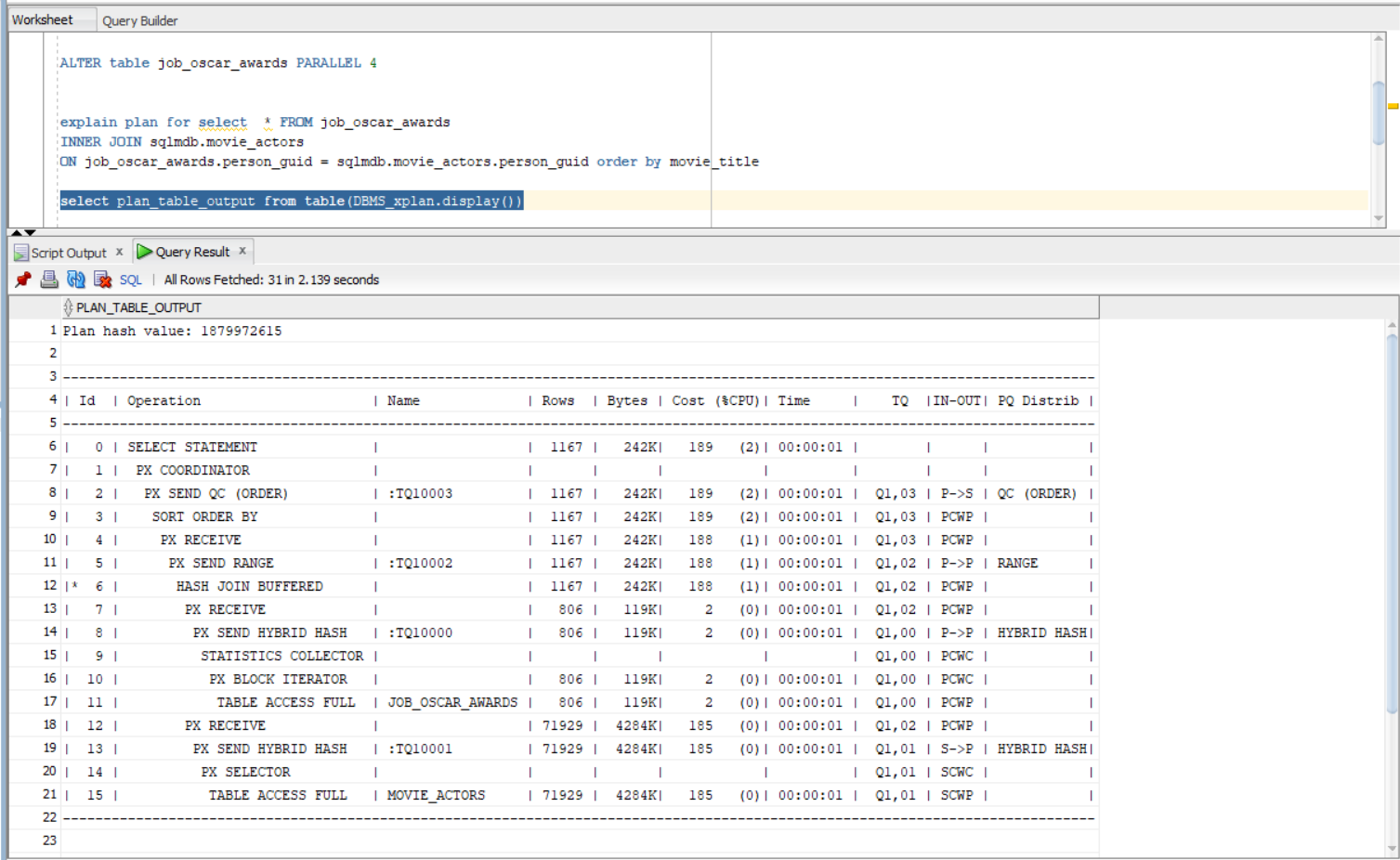
Parallel execution divides the task of implementing an SQL statement into multiple small units, each of which is executed by a separate process. Parallel execution is designed to effectively use multiple CPUs and disks to answer queries quickly. When multiple users use parallel execution at the same time, it is easy to quickly exhaust available CPU, memory, and disk resources.

**Degree Of Parallelism**

The number of parallel execution servers associated with a single operation is known as the degree of parallelism.

**Graphical user interface, text, application, email

Description automatically generated**

****

**Data Visualization**

Visualization of the data that was inserted is also an important aspect when it comes to understanding the data more efficiently. In this project visualizations are done for the Award category, Number of actors, Award winners, Number of awards for movies, and top 30 most award winners in the ACTOR\_OSCAR\_AWARDS table.

Chart, bubble chart

Description automatically generated

**Chart

Description automatically generated**

In the JOB\_OSCAR\_AWARDS table visualizations are done for the Award category, Number of actors, Award category, Award winners, Number of awards for top 30 movies, and top 30 most award winners and their job.

Chart, bubble chart

Description automatically generated

Chart

Description automatically generated with medium confidence

**Conclusion**

Created two tables “Actors Oscar Award” and “Job Oscar Award” with reference to the movies and persons which are in the existing movies, movie\_actors and movie\_jobs tables which will display the data of the actors, writers, and directors who won and nominated for Oscar awards. Did a couple of experiments on the project for performance tuning with indexing, optimizer mode, and Parallel Execution. Used data visualization to understand better and get more insights on data. Thus achieved the goal to include the awards table in the database efficiently.

**References**

<https://www.w3schools.com/sql>

<https://lucid.app/>