Name: Sarvesh Kaushik LAB7

Instructor: Luke Papademas

**[ The planets table, with comments on 2 fields ]**

**DROP TABLE planets;**

**CREATE TABLE planets**

**(pl\_ID number,**

**pl\_name varchar2(20),**

**pl\_diameter number,**

**pl\_class varchar2(20),**

**pl\_mass number, -- mass in tons x 10^21**

**pl\_dist\_fr\_sun number, -- distance x 10^6 miles**

**pl\_num\_moons number,**

**pl\_rings char(3));**

**[ The planets2 table ]**

**DROP TABLE planets2;**

**CREATE TABLE planets2 (pl\_ID number, picture BLOB);**

**[ The planets3 table ]**

**DROP TABLE planets3;**

**CREATE TABLE planets3 (pl\_id number, essay CLOB);**

**[The planets4 table]**

**DROP TABLE planets4;**

**CREATE TABLE planets4**

**(pl\_id number, music BLOB**

**Populating the Tables:**

**INSERT INTO planets**

**Values (1, 'Mercury', 4879.4,'Terrestrial',0.364,36,0,'No');**

**INSERT INTO planets**

**Values (2, 'Venus', 12103.6,'Terrestrial',5.37,67.2,0,'No');**

**INSERT INTO planets**

**Values (3, 'Earth', 12742,'Terrestrial', 6.58, 93,1,'No');**

**INSERT INTO planets**

**Values (4, 'Mars', 6779,'Terrestrial',0.708, 141.6,2,'No');**

**INSERT INTO planets**

**Values (5, 'Jupiter', 139822,'Jovian',2093,483.8,67, 'Yes');**

**INSERT INTO planets**

**Values (6, 'Saturn', 116464,'Jovian',627,890.8,62,'Yes');**

**INSERT INTO planets**

**Values (7, 'Uranus', 50724,'Jovian',95.7,1784.8,27,'Yes');**

**INSERT INTO planets**

**Values (8, 'Neptune',49244,'Jovian',113,2793.1,14,'Yes');**

**INSERT INTO planets**

**Values (9, 'Pluto', 1400 , 'Terrestrial',0.0144,3647.2,5,'No');**

**INSERT INTO planets**

**Values (10, 'PlanetX',99999 ,'Unknown',99999,9999 ,99,'No');**

**commit;**

**INSERT INTO planets2 values(2, EMPTY\_BLOB());**

**INSERT INTO planets2 values(3, EMPTY\_BLOB());**

**INSERT INTO planets2 values(4, EMPTY\_BLOB());**

**INSERT INTO planets2 values(5, EMPTY\_BLOB());**

**INSERT INTO planets2 values(6, EMPTY\_BLOB());**

**INSERT INTO planets2 values(7, EMPTY\_BLOB());**

**INSERT INTO planets2 values(8, EMPTY\_BLOB());**

**INSERT INTO planets2 values(9, EMPTY\_BLOB());**

**INSERT INTO planets2 values(10, EMPTY\_BLOB());**

**commit;**

**INSERT INTO planets3 values(1, EMPTY\_CLOB());**

**INSERT INTO planets3 values(2, EMPTY\_CLOB());**

**INSERT INTO planets3 values(3, EMPTY\_CLOB());**

**INSERT INTO planets3 values(4, EMPTY\_CLOB());**

**INSERT INTO planets3 values(5, EMPTY\_CLOB());**

**INSERT INTO planets3 values(6, EMPTY\_CLOB());**

**INSERT INTO planets3 values(7, EMPTY\_CLOB());**

**INSERT INTO planets3 values(8, EMPTY\_CLOB());**

**INSERT INTO planets3 values(9, EMPTY\_CLOB());**

**INSERT INTO planets3 values(10,Empty\_CLOB());**

**commit;**

**INSERT INTO planets4 values(1, EMPTY\_BLOB());**

**INSERT INTO planets4 values(2, EMPTY\_BLOB());**

**INSERT INTO planets4 values(3, EMPTY\_BLOB());**

**INSERT INTO planets4 values(4, EMPTY\_BLOB());**

**INSERT INTO planets4 values(5, EMPTY\_BLOB());**

**INSERT INTO planets4 values(6, EMPTY\_BLOB());**

**INSERT INTO planets4 values(7, EMPTY\_BLOB());**

**INSERT INTO planets4 values(8, EMPTY\_BLOB());**

**INSERT INTO planets4 values(9, EMPTY\_BLOB());**

**INSERT INTO planets4 values(10,Empty\_BLOB());**

**commit;**

# Adding values in the BLOB field

Graphical user interface, application, Word

Description automatically generated

# Adding values in the CLOB Field

Graphical user interface, text, application, Word

Description automatically generated

# Loading the Music Files into Planet4

Graphical user interface, application, Word

Description automatically generated

# Perform the following join between the planets, planets2 and planets3 tables.

**SELECT pl1.pl\_ID, pl1.pl\_name, pl2.picture, pl3.essays**

**FROM planets pl1, planets2 pl2, planets3 pl3**

# **WHERE pl1.pl\_ID = pl2.pl\_ID and pl1.pl\_ID = pl3.pl\_ID;**

Graphical user interface

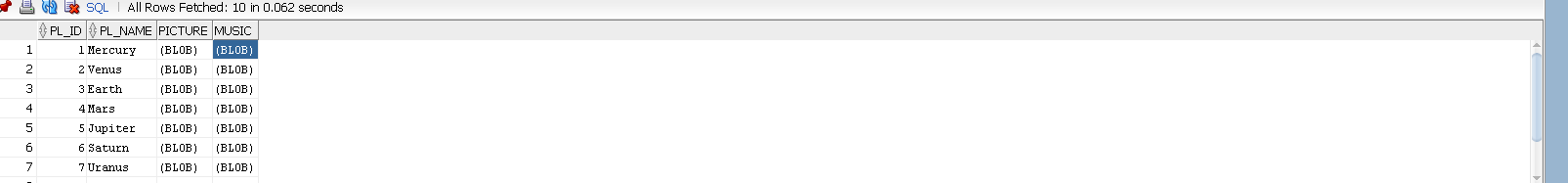
Description automatically generated with medium confidence

# Perform the following join between the planets, planets2 and planets4 tables.

SELECT pl1.pl\_ID, pl1.pl\_name, pl2.picture, pl4.music

FROM planets pl1, planets2 pl2, planets4 pl4

WHERE pl1.pl\_ID = pl2.pl\_ID and pl1.pl\_ID = pl4.pl\_ID;



# Create a Report on the Join

Graphical user interface, text, application, Word

Description automatically generated

# Insert a New Record for All Three Tables

INSERT INTO planets Values (11, 'SUN',1400,'Non-Terrestal',2093,483.8,67, 'Yes');

INSERT INTO planets Values (12, 'DwarfPlanet',2800,'Non Terrestal',2093,483.8,67, 'Yes');

INSERT INTO planets2 values (11, EMPTY\_BLOB());

INSERT INTO planets2 values (12, EMPTY\_BLOB());

INSERT INTO planets3 values (11, Empty\_CLOB());

INSERT INTO planets3 values (12, Empty\_CLOB());

Table

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

A picture containing table

Description automatically generated

Perform a Natural Join Between All Three Tables

SELECT \* from planets NATURAL JOIN planets2 NATURAL JOIN planets3.

A picture containing text

Description automatically generated

Take a snapshot of the report generated for the 3 - Table Join

Table

Description automatically generated with medium confidence

**Managers of media services that offer video and audio media files for viewing and listening, such as YouTube, Netflix, Amazon Prime, Hulu, and Google Play, have a tremendous task of storing the media, auditing the files, uploading new media and removing media. Customers are becoming more discriminatory, especially parents. How would you structure queries to filter media content that is geared toward children versus adults, comedies versus dramas and documentaries versus independent films? Give examples to support your answer.**

Managers for the media services is responsible for arranging the data according to the audiences. While creating the database schema, table needs to be normalized to remove the redundancy.

Category column could be introduced to distinguished between the adult and children centric content. Consider the following query,

For E.g.: Select \* from List where Category = “Children”;

Select \* from List where Category = “Adult”;

Moreover, you can filter by Genre by making a separate genre column and filtering the data based on that.

Select \* from List where Category = “Children” AND Genre = “Comedy”.

Moreover, with each field type one could consider adding a film category aa documentary, and Independent Movies.

**Many cultural institutions, such as art museums, planetariums, orchestras, and zoos, also maintain databases containing files (Customer / Member Information, Media Files, Image Files (photographs), (digital news articles) that also must be maintained. Construct a database containing various tables that will classify such files (as mentioned here) for Century City’s 21st Century exhibition on the history of the city. Be descriptive as to the attributes of each table.**

Consider the following most important tables in the Century City 21st exhibition. The Attend info will consist of information about all the event attendees, it will be responsible for storing the basic information of the attendees such as address, name arrival time and location.

Digital info table will primarily focus on storing all the digital info, legal documents related to the event. It’s a data repository where event coordinators will store all the important digital files, So it can be retrieved for the later use.

Ticket Info: Ticket info will consist of an important information related to the ticket sales such as Ticket Type, number of copies sold, and copies left.

One of the important tables could be Staff table, it is important to have right team to coordinate effectively and make event successful. Data stored in the staff table will be helpful coordinating with individual team’s members based on their role.

**Graphical user interface, application

Description automatically generated**

**Intellectual Property includes patents, trademarks, copyrights, and trade secrets. Perform research as to the legal lifetime of these four important intellectual properties in the United States.**

Intellectual Properties: Authorities are responsible protection of Intellectual Property rights across globe. Intellectual Property arises when company delivers innovative product or services. The unique selling proposition of an organization revolves around the companies Innovative Capacity.

The organization innovative growth rate determines its IP worth in the current market. The intellectual property enforcement office is responsible for preserving rights on individual products and services.

There are different types of Intellectual Properties such as Patents, Trademarks, Copyrights, and trade secrets.

**Patents**: A patent gives powerful right to the owner or product or service to prohibit others from using their blueprint or formula without signing mutual agreement. The rules, regulations and laws oriented towards patent may differ from country to country.

Under the current law, the patent is having a strict timeline of the 20 years from the patent filing date. It remains active as a strong property of an organization.

**Trademarks:** Trademarks are used to represent the brand of the company. Trademarks are the symbols or set of words that differentiates companies product and services from other competitors. Trademarks are beneficial in creating your brand. The trademark needs to be renewed every 10 years to keep the legal existence of the company.

**Copyrights:** Copyright helps to establish the ownership of the work for an individual or an organization. The copyrighted work could include media, musical content, web series, movies, graphiti, images, digital art, Comicon characters and so on. A typical copyright lasts for more than 70 years after the death of the author.

**Trade Secrets:** Organizations possess the valuable information and the data which can be traded under as an Intellectual Property and is often known as a Trade Secret. There is certain regulation according to the trade secrets treaty, as receiving party must agree to keep the information confidential at any cost. There are no explicit costs associated with registering the trade secrets. Once the trade secret protections have been established, it lasts till it is kept hidden from external entities.

**When conducting research in astronomy, there are often new discoveries made that may add to, corroborate, or contradict known data. You are the database manager for NASA. How are you do deal with these discoveries and their placement into the appropriate files?**

As a database manager for NASA, one most important responsibility would be to make the applications scalable due to wide variety of the data. As the new discoveries unlocks ensuring the execution of proper procedures to update the data in the database table would be necessary.

The data related to the new discoveries should be structured data and must be stored in terms of log files. The database should be normalized to reduce the database redundancy eliminating the updating, insertions, and deletion anomalies. The database integrity would be one of the most important factors to consider while creating relationships between two or more databases**.**

**Would it be preferable to store astronomical data in NoSQL files or in traditional SQL files? Support your answer.**

As a NASA database manager one of the important factors would be the scalability of the data as new discoveries needs to be stored. The efficient database structure to achieve the maximum scalability would be NoSQL. It supports the scale out architecture, it can store the large volumes of data over the large clusters of machines. More machines can be added to the cluster to achieve more scalability.

The data can be stored in the NoSQL database in easier to read and retrieve format as compared to SQL which supports only structured data stored in terms of table, rows, and columns. It supports the core functionality of the Big Data well with graph and network interrelated data.

In the NoSQL database the data is stored in the Key-Value Pairs, New columns can be simply added without disrupting the existing data structure. The Astronomical data is very dynamic and requires higher scalability, it is more volatile and subject to change. The NoSQL database is a perfect fit for rapidly changing, unstructured large volume of data.