ITMD 523 *Advanced Topics in Data Management* HW 4

Student Name **Sivaranjani Prabasankar** Section

Instructor ***Luke Papademas*** Due Date

Part	1	2	3	4	TOTAL	Score
Maximum Points	25 points	25 points	25 points	25 points	**100** points	

Textbook Reading Assignment

Thoroughly read Week 1 - 5 course lecture notes.

Part 1 Concepts, Topics, Glossary Terms - Advanced Topics in Data Management

Comment and expound, in detail, on each of the following questions.
Use examples to support your answers and indicate when and / or where the individual concepts would apply.

(1) (Fat Clients)

Envision a Club Membership Tracking database application. How would a fat client approach be used to implement such an application?

A thick client (also called a fat client) is one that will perform the bulk of the processing in client/server applications. With thick clients, there is no need for continuous server communications as it is mainly communicating archival storage information to the server. As in the case of a thin client, the term is often used to refer to software, but again is also used to describe the networked computer itself.
If your applications require multimedia components or that are bandwidth intensive, you'll also want to consider going with thick clients. One of the biggest advantages of thick client rests some operating systems and software being unable to run on thin clients. Thick clients can handle these as it has its own resources.
A fat client has several advantages, including the following:

- Fewer server requirements because it does most of the application processing
- More offline work because a server connection is often not required
- Multimedia-rich application processing, such as video gaming facilitation, because there are no increased server bandwidth requirements
- Runs more applications because many fat clients require that an operating system reside on a local computer
- Easy network connection at no extra cost because many users have fast local PCs
- Higher server capacity because each fat client handles more processing, allowing the server to serve more clients

Example: Club Membership Tracking Database
A club consists of many amenities available for its hierarchy of members.

- Facilities include SPA, swimming pool, outdoor games like cricket, Basketball, Golf, Theater, GYM, Indoor games like tennis, badminton, children's park, restaurant, grocery store, yoga class, Salsa class, karate etc.
- All these facilities are to be managed by the management.
- Access to these facilities depends on the membership type which has been stored in database. Let us consider that there are 4 types membership that anyone can avail: Platinum, Gold, silver and Bronze membership.
  - ⇒ Platinum members are allowed at all departments for unlimited no. of. hours
  - ⇒ Gold members are allowed at all departments for limited hours.
  - ⇒ Silver member can avail all the sports section along with GYM and SPA.

⇒ Bronze member is only allowed to avail SPA, GYM and swimming pool

All these details are to be fed in the database and maintained.
The database is accessible locally as the establishment of the Club is only at a certain place.
All the details for example membership details, hours of usage, member allowance, facilities maintenance, changes in the club, club policies, club employees, club employee work shift details etc. are all to be maintained only in the premises of the club.
None of the other actors like members must access the data in this scenario. Since the club area is big, even though database is local it will to be networked across.

(2) (Location Transparency)

Location transparency is a property of a distributed database management system (DDBMS) . Here database access requires the user to know only the name and not the location of the data items.

Explain how a typical search engine application incorporates location transparency.

Distribution transparency is the property of distributed databases by the virtue of which the internal details of the distribution are hidden from the users.

Location transparency

- It Enables resources to be accessed **without knowledge of their physical or network location**.
- It is the middle level of distribution transparency.
- It ensures that the user can query on any table(s) or fragment(s) of a table as if they were stored locally in the user's site.
- To achieve location independence, a dictionary containing locations of all data is needed.

The fact that the table or its fragments are stored at remote site in the distributed database system, should be completely oblivious to the end user. The address of the remote site(s) and the access mechanisms are completely hidden.

To incorporate location transparency, DDBMS should have access to updated and accurate data dictionary and DDBMS directory which contains the details of locations of data.

Example: TABLES in Distributed Database.

(3) (Artificial Key)

The types of database key fields include artificial keys, natural keys and surrogate keys. Is an artificial key hidden to the users? Provide some examples of artificial keys.

When primary key is very large and complex, then 'Artificial keys or Surrogate Key' is used. It does not exist in the external reality but can be verified for syntax or check digits inside itself.

- They are allowed when no property has the parameter of primary key.
- They are generally not used in professional organization.
- Artificial keys are artificially generated key and It will not have meaning to the table.

There are few properties of artificial keys.

⇒ They are unique because these just created when you don't have any natural primary key.

⇒ They are integer values.

⇒ One cannot find the meaning of surrogate keys in the table.

⇒ End users cannot create artificial key.

⇒ It signed by DBMS each time when a row is created, and the values never change.

⇒ They are often hidden from the results and reports

Artificial keys are allowed when

- No property has the parameter of primary key.
- The primary key is huge and complex.

Example: In the grocery business, this might be where you find bread baked in the store, rather than bread bought from a national bakery chain. The check digits still work the same way to validate the barcodes, but you must verify them inside your own enterprise. It's up to the DBA to maintain a trusted source for them.

Another technique is to take a natural key that is made up of many known attributes and put them into a formula to generate a key -- a hashing function. There are several downsides to this approach. The hashing function must be perfect (that means no collisions or duplicate values from different inputs), and you still must guarantee the uniqueness of the subset of attributes that make up the natural key. You will sometimes see an artificial key called a "surrogate key,". If you must construct a key yourself, it takes time to design them, to invent a validation rule, set up audit trails, etc. It's not easy work

(4) (Two - Phase Commit)

The two - phase commit protocol is a distributed process which lets all sites in a distributed system agree to commit a transaction or rollback a transaction.

Consider an online retail store Web application. Explain how a two - phase commit protocol could be used when entering data for corporate salesperson expense reports.

Distributed two-phase commit reduces the vulnerability of one-phase commit protocols. The steps performed in the two phases are as follows with an example from **Online Purchase i.e. Transfer money from User account to Amazon Account** to confirm payment

Phase 1: Prepare Phase

- After each slave has locally completed its transaction, it sends a "DONE" message to the controlling site. When the controlling site has received "DONE" message from all slaves, it sends a "Prepare" message to the slaves. – Debit at User, credit at Amazon, tell the client "okay"

- The slaves vote on whether they still want to commit or not. If a slave wants to commit, it sends a "Ready" message. Banks will check for existing balance and amount to be transferred, sends READY message if there are no conflicts.
- A slave that does not want to commit sends a "Not Ready" message. This may happen when the slave has conflicting concurrent transactions or there is a timeout. Banks will check for existing balance and amount to be transferred, sends Not READY message in case of low balance

Phase 2: Commit/Abort Phase

- After the controlling site has received "Ready" message from all the slaves Require both banks to do it, or neither –
  - The controlling site sends a "Global Commit" message to the slaves.
  - The slaves apply the transaction and send a "Commit ACK" message to the controlling site. Perform actions – Debit and Credit
  - When the controlling site receives "Commit ACK" message from all the slaves, it considers the transaction as committed. Message receipt to User and Amazon.
- After the controlling site has received the first "Not Ready" message from any slave –
  - The controlling site sends a "Global Abort" message to the slaves.
  - The slaves abort the transaction and send a "Abort ACK" message to the controlling site. Cancel actions – Debit and Credit
  - When the controlling site receives "Abort ACK" message from all the slaves, it considers the transaction as aborted. Message receipt to User and Amazon.

(5) (Fact Tables)

In data warehouse design, fact tables are used in the dimensional model whereby such tables are found at the center of a star schema bounded by dimension tables. Fact tables consist of facts pertaining to a business process, such as sales expenses by year by service.

Consider a fact table that deals with student grades and averages. What would be some measures (fields) that the fact table would contain?

The fact table is at the core of the star schema. Fact table stores facts or measure of interests. Normally facts are numbers that can be aggregated, summarized or rolled up.

Thus, the fact table consists of two types of columns. The fact table contains surrogate keys as a part of its primary key. Those keys are the foreign key of the dimension tables. These foreign keys column allows joins with dimension tables, and the measures columns contain the data that is being analyzed.

Student Name **Sivaranjani Prabasankar** Section

Star schema is a dimensional design for a relational database often used in a data warehouse system. The star schema has a fact table at the center, and the fact table is surrounded by a number of dimension tables. The star schema name comes from the appearance of the schema, which looks like a star.

In the star schema, related dimensions grouped together as columns in dimension tables and used to store the context of the facts stored in the fact table.

Consider the following example: A fact table may look like this:

Stud_ID	Subject_ID	Grade
101	ITMD523	A
102	ITMD525	A
103	ITMD527	A

A dimension Tables

Subject ID	Sub_Name	Credits
ITMD523	Data Management	3
ITMD525	Data Mining	3
ITMD527	Data Analytics	3

Stud_ID	Stud_Name	Marks
101	Anusha	90
102	Megha	92
103	Shiv	91

So, when we combine the three tables we can come to several conclusions like "The student Anusha scored an A in Data Management Whose Course Id is ITMD 523 with 3 credits". Thus, a fact table and dimensional table give an order to any database and help in better database performance as the records are well organized.

Part 2 DBMS Concepts - Advanced Topics in Data Management

(1) (DDL and DML Operations)

Consider the Oracle table structures which follows. The tables store department and employee information. Then, answer the following exercises, which are based on these tables.

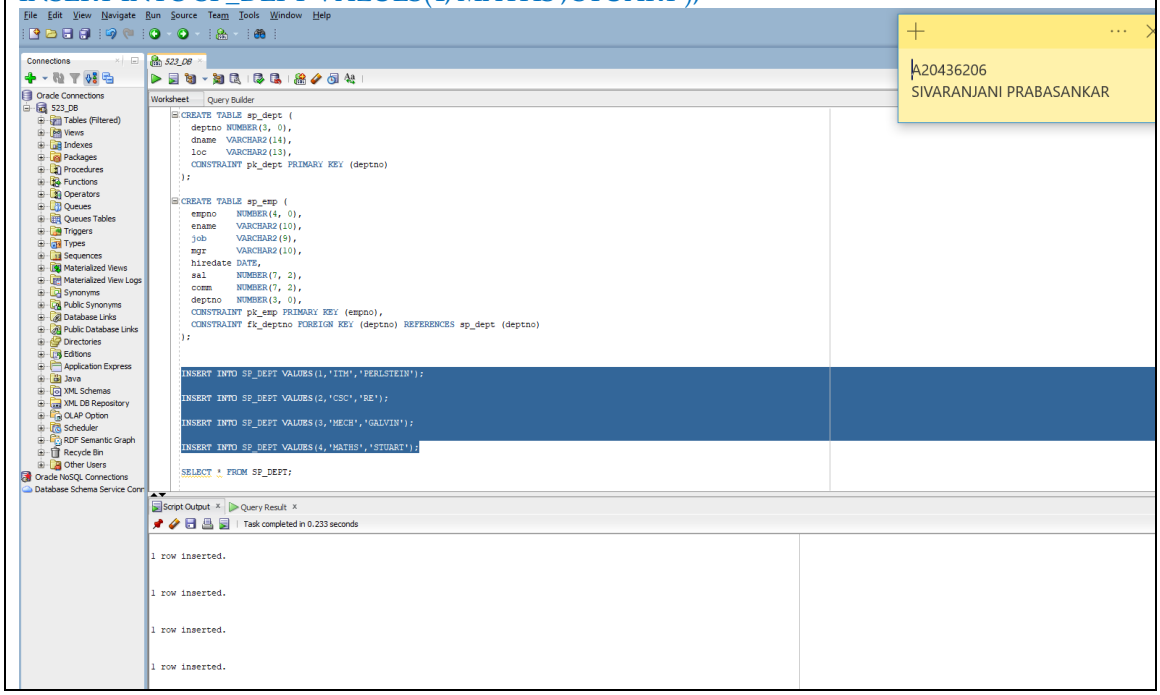
```
--Table Departments:
CREATE TABLE dept (
  deptno NUMBER(3, 0),
  dname  VARCHAR2(14),
  loc    VARCHAR2(13),
  CONSTRAINT pk_dept PRIMARY KEY (deptno)
);
--Table Employees:
```

Student Name **Sivaranjani Prabasankar** Section _____

```
CREATE TABLE emp (
  empno    NUMBER(4, 0),
  ename    VARCHAR2(10),
  job      VARCHAR2(9),
  mgr      VARCHAR2(10),
  hiredate DATE,
  sal      NUMBER(7, 2),
  comm     NUMBER(7, 2),
  deptno   NUMBER(3, 0),
  CONSTRAINT pk_emp PRIMARY KEY (empno),
  CONSTRAINT fk_deptno FOREIGN KEY (deptno) REFERENCES dept (deptno));
```

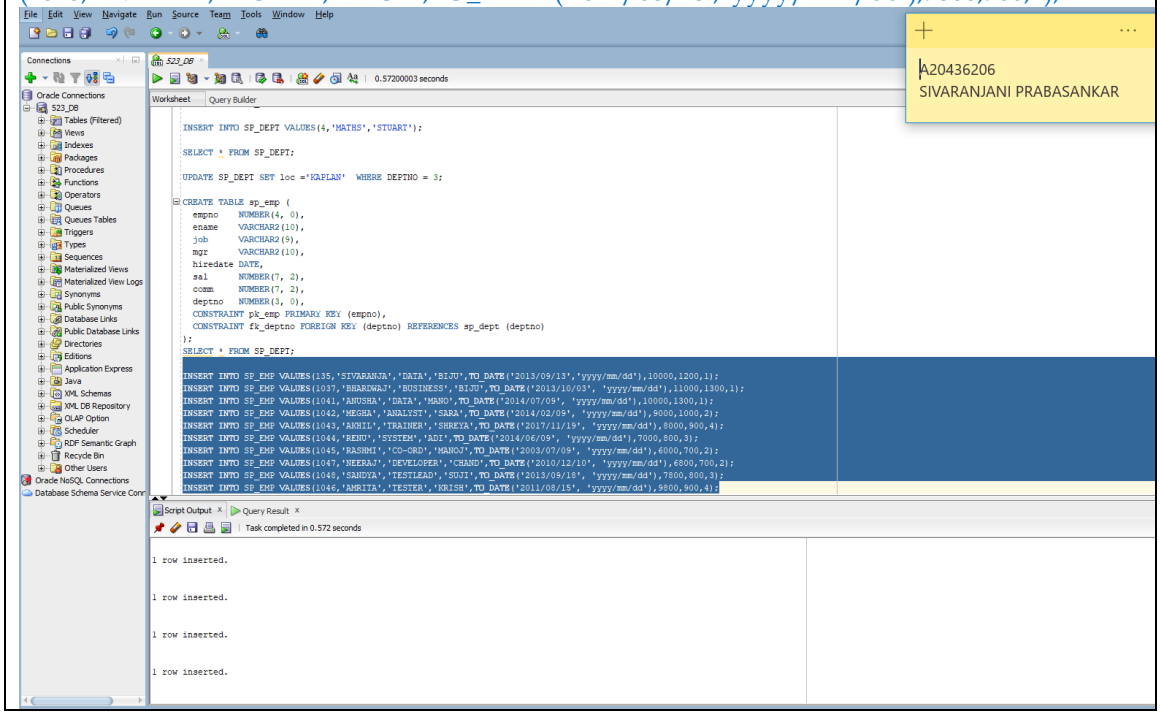
(a) Insert at least <u>four</u> records in department " dept " table.
Use fictitious but realistic data to populate the table.

```
INSERT INTO SP_DEPT VALUES(1,'ITM','PERLSTEIN');
INSERT INTO SP_DEPT VALUES(2,'CSC','RE');
INSERT INTO SP_DEPT VALUES(3,'MECH','GALVIN');
INSERT INTO SP_DEPT VALUES(4,'MATHS','STUART');
```



(b) Insert at least <u>six</u> records in employees " emp " table and ensure that the primary key / foreign key reference is maintained. Use your own name and location for one of the records. When inserting the rows into the emp table use the `to_date()` function to cast a string literal into an Oracle (or similar RDBMS) DATE format.

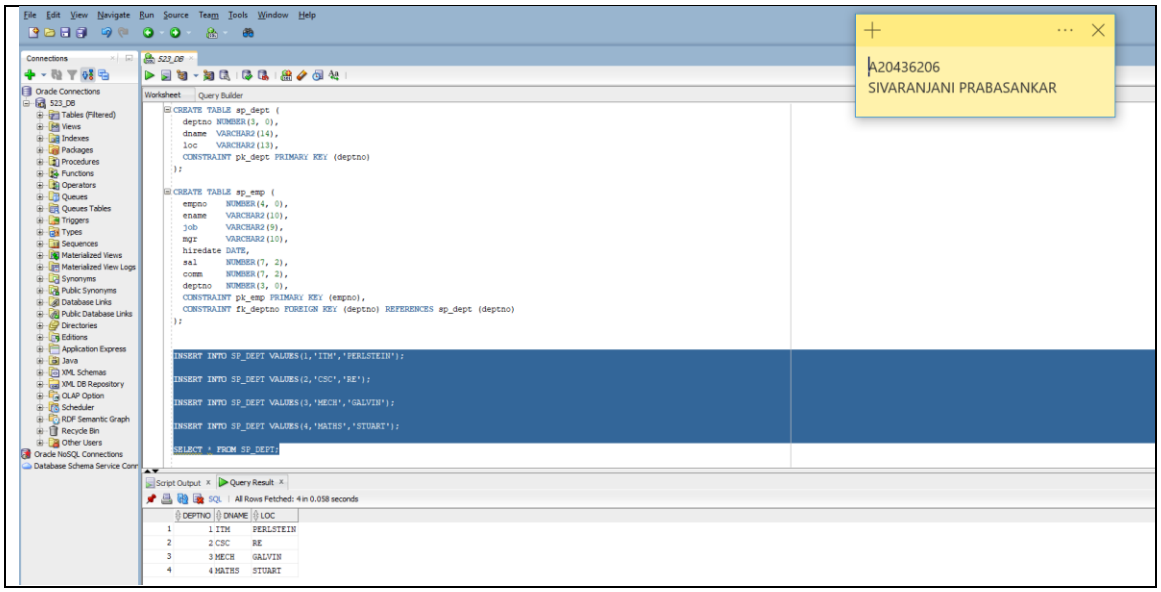
```
INSERT INTO SP_EMP VALUES
(135,'SIVARANJA','DATA','BIJU',TO_DATE('2013/09/13','yyyy/mm/dd'),10000,1200,1);
INSERT INTO SP_EMP VALUES
(1037,'BHARDWAJ','BUSINESS','BIJU',TO_DATE('2013/10/03', 'yyyy/mm/dd'),11000,1300,1);
INSERT INTO SP_EMP VALUES
(1041,'ANUSHA','DATA','MANO',TO_DATE('2014/07/09', 'yyyy/mm/dd'),10000,1300,1);
INSERT INTO SP_EMP VALUES
(1042,'MEGHA','ANALYST','SARA',TO_DATE('2014/02/09', 'yyyy/mm/dd'),9000,1000,2);
INSERT INTO SP_EMP VALUES
(1043,'AKHIL','TRAINER','SHREYA',TO_DATE('2017/11/19', 'yyyy/mm/dd'),8000,900,4);
INSERT INTO SP_EMP VALUES
(1044,'RENU','SYSTEM','ADI',TO_DATE('2014/06/09', 'yyyy/mm/dd'),7000,800,3);
INSERT INTO SP_EMP VALUES
(1045,'RASHMI','CO-ORD','MANOJ',TO_DATE('2003/07/09', 'yyyy/mm/dd'),6000,700,2);
INSERT INTO SP_EMP VALUES
(1047,'NEERAJ','DEVELOPER','CHAND',TO_DATE('2010/12/10', 'yyyy/mm/dd'),6800,700,2);
INSERT INTO SP_EMP VALUES
(1048,'SANDYA','TESTLEAD','SUJI',TO_DATE('2013/09/18', 'yyyy/mm/dd'),7800,800,3);
INSERT INTO SP_EMP VALUES
(1046,'AMRITA','TESTER','KRISH',TO_DATE('2011/08/15', 'yyyy/mm/dd'),9800,900,4);
```



(c) Display all data in the department " dept " table.

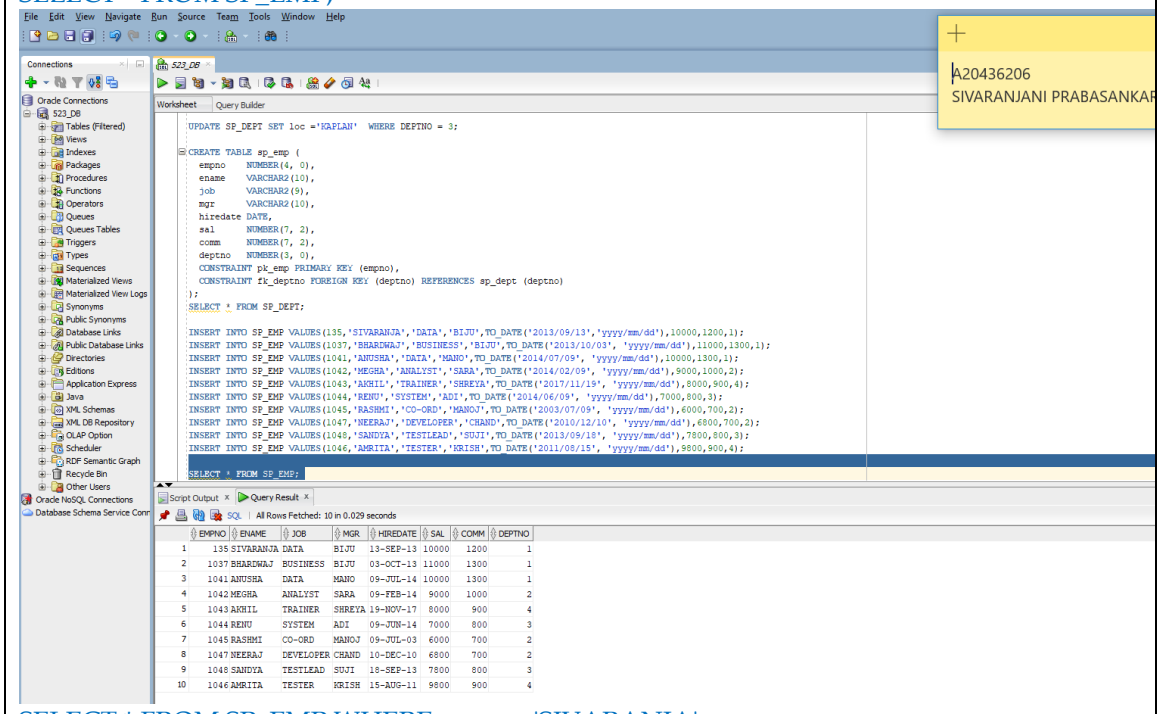
```
SELECT * FROM SP_DEPT;
```

Student Name **Sivaranjani Prabasankar** Section



(d) Display only your own record in the employees " emp " table.

SELECT * FROM SP_EMP;



SELECT * FROM SP_EMP WHERE ename='SIVARANJA';

```
INSERT INTO SP_EMP VALUES(1037,'BHARDWAJ','BUSINESS',
INSERT INTO SP_EMP VALUES(1041,'ANUSHA','DATA','MANO',
INSERT INTO SP_EMP VALUES(1042,'MEGHA','ANALYST','SARA
INSERT INTO SP_EMP VALUES(1043,'AKHIL','TRAINER','SHRE
INSERT INTO SP_EMP VALUES(1044,'RENU','SYSTEM','ADI',T
INSERT INTO SP_EMP VALUES(1045,'RASHMI','CO-ORD','MANO
INSERT INTO SP_EMP VALUES(1047,'NEERAJ','DEVELOPER','C
INSERT INTO SP_EMP VALUES(1048,'SANDYA','TESTLEAD','SU
INSERT INTO SP_EMP VALUES(1046,'AMRITA','TESTER','KRISH',TO_DATE('2011/08/15', 'yyyy/mm/dd'),9800,900,4

SELECT * FROM SP_EMP WHERE ename='SIVARANJA';
DELETE FROM SP_EMP WHERE DEPTNO=4;
```

A20436206
SIVARANJANI PRABASANKAR

Script Output | Query Result

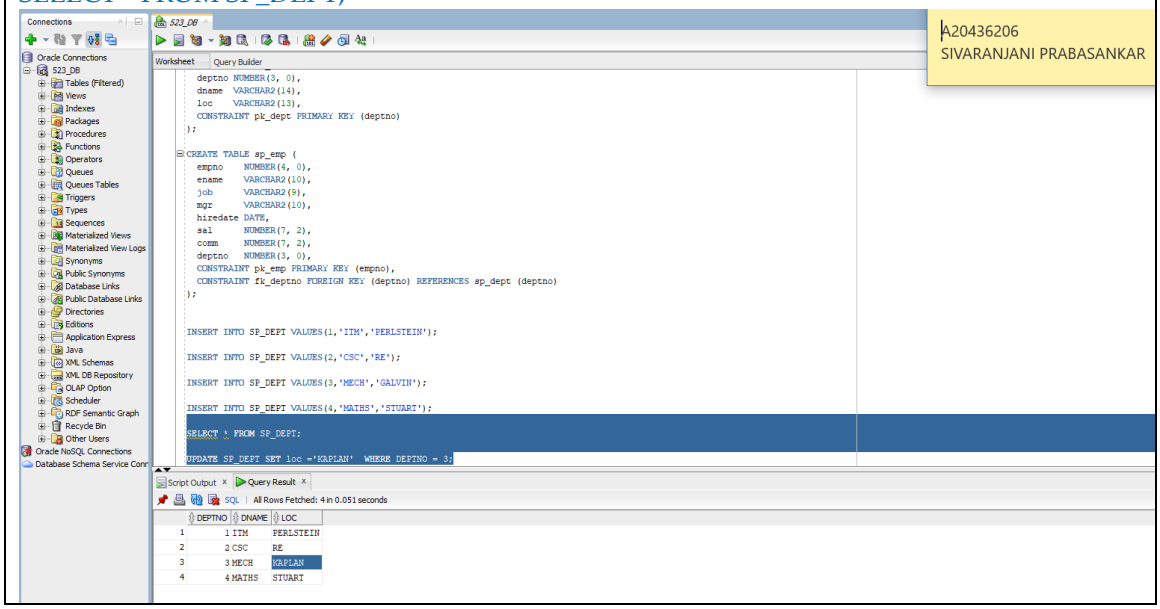
SQL | All Rows Fetched: 1 in 0.018 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	135	SIVARANJA	DATA	BIJU	13-SEP-13	10000	1200	1

ey down to perform "Go to Declaration"

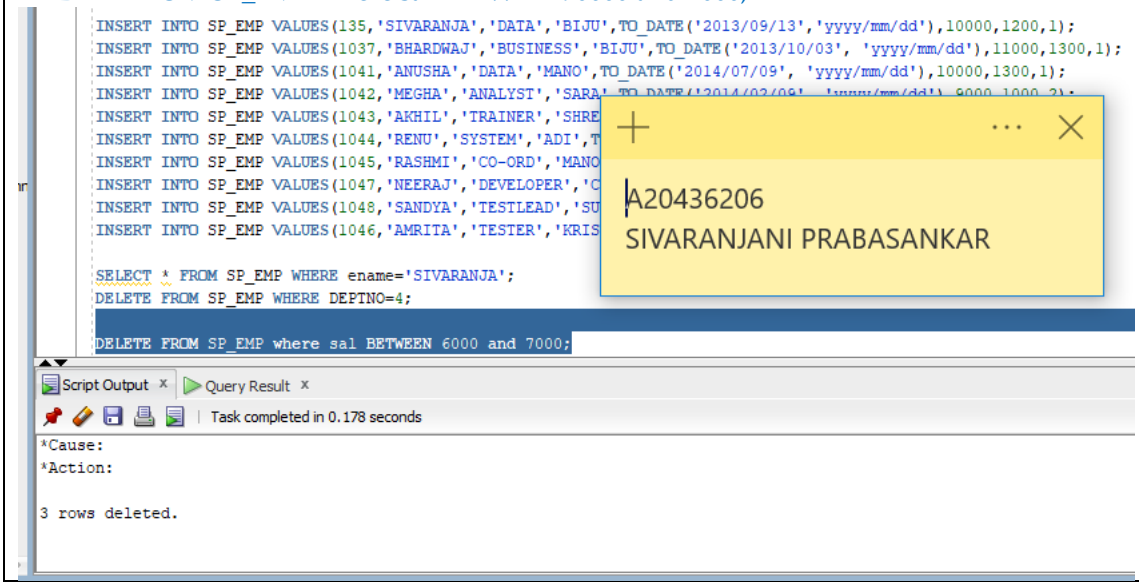
(e) Update an address or location of at least one department.

```
UPDATE SP_DEPT SET loc ='KAPLAN'  WHERE DEPTNO = 3;
SELECT * FROM SP_DEPT;
```



(f) Delete the record(s) of at least one employee.

DELETE FROM SP_EMP where sal BETWEEN 6000 and 7000;



(g) Can you delete a department that still has employee? Explain your answer.

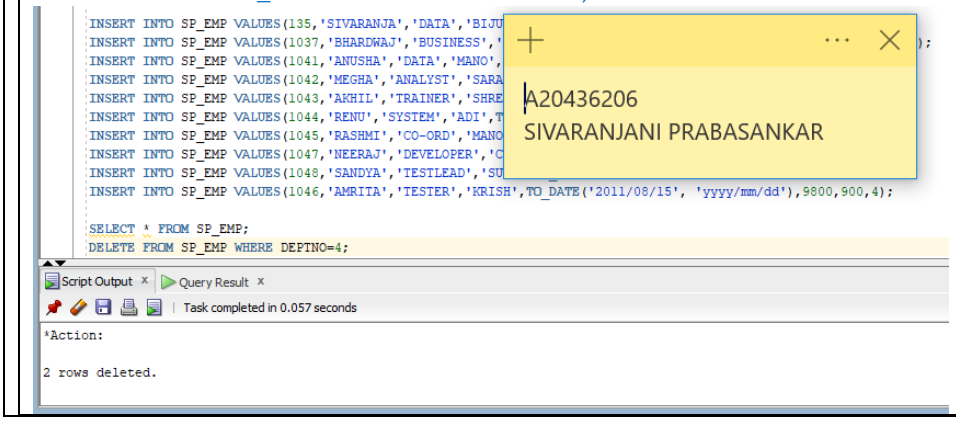
No, we can't delete due to referential integrity property of SQL. This property exists in Database to avoid orphaned records.

DELETE FROM SP_DEPT WHERE deptno=1;

Error starting at line : 52 in command -
DELETE FROM SP_DEPT WHERE deptno=1
Error report -
ORA-02292: integrity constraint (ORA_SPRABASANKAR.FK_DEPTNO) violated - child record found

(h) Can you delete an employee who works for a department? Explain your answer.

Yes, we can delete and the query is
DELETE FROM SP_EMP WHERE DEPTNO=4;



Student Name **Sivaranjani Prabasankar** Section _____

(2) (DDL and DML Operations)

(a) Construct a simple natural join between the DEPT and EMP tables based on the primary key of the DEPT table (DEPTNO) and the DEPTNO foreign key in the EMP table.

SELECT * FROM sp_dept D NATURAL JOIN sp_emp E;

```
SELECT E.EMPNO,E.ENAME,E.JOB,d.deptno,d.dname FROM sp_dept D JOIN on sp_emp E where d.deptno = e.deptno;

SELECT * FROM sp_dept D NATURAL JOIN sp_emp E;

select D.DNAME, COUNT(E.EMPNO) AS EMPCOUNT from SP_DEPT D INNER JOIN  SP_EMP E ON D.DEPTNO=E.DEPTNO GROUP BY(D.I
```

A20436206
SIVARANJANI PRABASANKAR

Script Output × Query Result ×

SQL | All Rows Fetched: 5 in 0.016 seconds

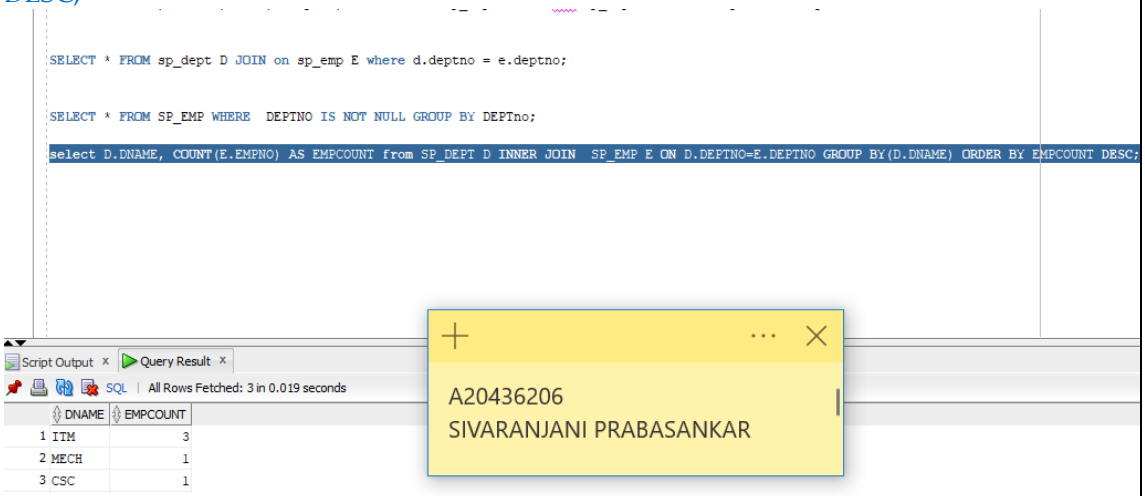
	DEPTNO	DNAME	LOC	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
1	1	ITM	PERLSTEIN	135	SIVARANJA	DATA	BIJU	13-SEP-13	10000	1200
2	1	ITM	PERLSTEIN	1037	BHARDWAJ	BUSINESS	BIJU	03-OCT-13	11000	1300
3	1	ITM	PERLSTEIN	1041	ANUSHA	DATA	MANO	09-JUL-14	10000	1300
4	2	CSC	RE	1042	MEGHA	ANALYST	SARA	09-FEB-14	9000	1000
5	3	MECH	KAPLAN	1048	SANDYA	TESTLEAD	SUJI	18-SEP-13	7800	800

(b) Similar to the above exercise, construct a join between the DEPT and EMP tables based on these specifications:

- display records from both tables
- display the department name and the count of all of the employees
- base the join on the primary key of the DEPT table (DEPTNO) and the DEPTNO foreign key in the EMP table
- group the records by the department name and order the records by one of the columns, in descending order

- display records from both tables include attributes from both table
- display the department name and the count of all of the employees using COUNT function
- base the join on the primary key of the DEPT table (DEPTNO) and the DEPTNO foreign key in the EMP table using INNER JOIN
- group the records by the department name and order the records by one of the columns, in descending order using DESC

select D.DNAME, COUNT(E.EMPNO) AS EMPCOUNT from SP_DEPT D INNER JOIN SP_EMP E ON D.DEPTNO=E.DEPTNO GROUP BY(D.DNAME) ORDER BY EMPCOUNT DESC;



Part 3 Data Modeling Concepts - Advanced Topics in Data Management

(1) (Database Table Normalization)

Examine this structure for the Projects table, which shows the project number and name and the team assigned to the project. The company bills its clients based on the hours spent on each project. The hourly billing rate depends on the employee's position.

```
Projects
(
      Project Number, Project Name, Team Number, Team Name,
      Job Classification, Hourly Billing Rate, Number of Hours
)
```

Using 2NL normalization techniques, split the above Projects table into three separate tables, given below. Decide how the original fields from the Projects table will be allocated amongst the three tables.

Table PROJECTS (. . .)

Table TEAMS (. . .)

Table ASSIGNMENTS (. . .)

For a table to be in **2nd Normal Form**, all the following conditions must be satisfied:

a. Table should be in 1st Normal Form.

b. There should not be any partial dependency on the primary key. Or it can be re-stated as all the non – key attributes must be dependent on the primary key.

1ST NORMAL FORM

Projects (Project Number, Project Name, Team Number, Team Name, Job Classification, Hourly Billing Rate, Number of Hours)

2ND NORMAL FORM

PROJECTS (Project Number (PK), Project Name)

Project Number will be the primary key for table PROJECT

TEAMS (Team Number (PK), Team Name, Project Number (FK))

Team Number will be the primary key for table TEAM whereas Project Number will act as foreign key links with table PROJECT

ASSIGNMENTS (Project Number (PK), Job Classification (PK), Hourly Billing Rate, Number of Hours)

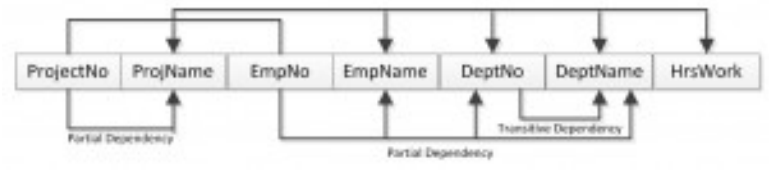
Project Number and Job Classification together act as primary key for table ASSIGNMENTS

The company bills its clients based on the hours spent on each project. ➔ This can be calculated by using **Project Number and Number of Hours in ASSIGNMENTS table**

The hourly billing rate depends on the employee's position **Job Classification and Hourly Billing Rate in ASSIGNMENTS table**

(2) Describe a dependency diagram and explain its purpose.

A dependency diagram illustrates the various dependencies that might exist in a non-normalized table. A non-normalized table is one that has data redundancy in it.



The following dependencies are identified in this table:

ProjectNo and EmpNo, combined, are the PK.

Partial Dependencies:

ProjectNo —> ProjName

EmpNo —> EmpName, DeptNo,

ProjectNo, EmpNo —> HrsWork

Transitive Dependency:

DeptNo —> DeptName

The left side of the above FD diagram is called the determinant, and the right side is the dependent.

Partial dependency

A Partial dependency means that a nonprime attribute is functionally dependent on part of a candidate key. (A nonprime attribute is an attribute that's not part of any candidate key.)

Student Name **Sivaranjani Prabasankar** Section

Transitive Dependency A functional dependency is said to be transitive if it is indirectly formed by two functional dependencies. For e.g. Book → Author Name → Author Age
X -> Z is a transitive dependency if the following three functional dependencies hold true:
X->Y, Y does not ->X, Y->Z

Functional Dependencies
A functional dependency (FD) is a relationship between two attributes, typically between the PK and other non-key attributes within a table. For any relation R, attribute Y is functionally dependent on attribute X (usually the PK), if for every valid instance of X, that value of X uniquely determines the value of Y. This relationship is indicated by the representation:

X – – – -> Y

Part 4 Data Design Concepts - Advanced Topics in Data Management

(1) (Big Data)

https://www.oracle.com/big-data/index.html

Review and study any information at the above link and view any videos that are posted. Then select a role of your choice from the list below.

- Chief Data Officer (CDO)
- Chief Information Officer (CIO)
- Enterprise Architect
- Data Scientist
- Business Leader

Write a brief essay that explains how your role can benefit or can impede from Oracle's Big Data applications.

As a Data Scientist if

1) I have **challenges in deploying models into production**, I would use the open-source R environment within the Oracle Database, Hadoop, and business intelligence to apply massive scalability, performance, and geospatial analysis from the lab to production.

2) My **slow analytical performance limit time** to model, I would mine data easier to create more value from available data on daily basis to detect patterns within large amounts of data and act fast.

3) I **struggle to communicate about data science with business managers**, I can Demonstrate the value of big data and analytics to go beyond reactive intelligence to proactive insight now of each customer contact. As advanced analytics enable you to contribute to business performance.

(2) (Data Science)

Visit the following link to examine the IBM Data Science Experience.

Student Name **Sivaranjani Prabasankar** Section

http://www.ibm.com/analytics/us/en/technology/cloud-data-services/dataworks/

Read any text and view any videos at the above link that discuss the realm of Data Science is an environment that has everything a data scientist needs to be successful.

Write a brief essay that comments on Data Science and how useful it can be for database Cloud operations.

Need of Data Science in Cloud

4) **Need to run scalable data science:**

Now-a-days, not only the number of people using Internet has increased but the amount of data increased exponentially. We are sticking with physical servers. Either we buy a new, more powerful server or load this one to maximum (which would have again run out of juice). The last thing you want is that your data scientists are staring at the screen waiting for data to be processed. A machine on cloud can scale on a click of the button, without much hassle. So, your current scripts and models can easily run when the data behind the model grows multiple times.

2) Cost: While scalability is one benefit, cost is another. If you need a higher computational infrastructure like for mining social media data for an annual event insight for real time. We can't really buy a new machine for that. The costs will be prohibitive. Simple, rent out a higher configuration for a few hours or days and you have solution to your problem at a fraction of the cost.

3) Collaboration: If we want to work simultaneously along with several data scientists and we don't want everyone to create a copy of the data and code in their local machines.

4) Sharing: Larger ecosystem for machine learning system deployments: A few cloud services like AWS, Azure provide complete ecosystem to collect data, run your models and then deploy them. In case of physical machine, you will need to set this up yourself.

5) Use for building quick prototypes: It is much easier to use the out of the box services on the cloud. You can quickly build prototypes without worrying about versions and scalability. Once you have proven the concept, you can always build a production stack later.

Challenges in running data science on cloud:

While cloud computing comes with its own benefits, there are a few challenges as well.

1) **Reluctance to share data with a third party:** From security perspective it is reluctant to share data on cloud outside of the company. This is at times driven by regulatory requirements or legal guidance, but at times the reasons are irrational as well. For example, quite a few banks are not comfortable uploading their data on cloud for purpose of analysis.

2) **Need to upload / download huge amount of data:** In case we have huge amount of data in our data center – a onetime upload of this data might be a huge challenge, if our internet infrastructure is not robust.