Business Intelligence and Applications

Q1. a) What are the hardware and software requirements to build group decisions support system?

b) How is knowledge represented in knowledge base of an expert system?

c) What are the features of decision support system?

Ans. A Decision Support System (DSS) is an interactive computer-based system or subsystem intended to help decision makers use communications technologies, data, documents, knowledge and/or models to identify and solve problems, complete decision process tasks, and make decisions.

 Decision Support Systems, conventionally called as DSS deal with the design and the use of cognitively compatible computerized systems for-

1. Assisting the managers in taking more effective decisions concerning semi-structured and unstructured tasks.
2. Supporting, rather than replacing, managerial intuition and judgement
3. Improving the effectiveness of decision making rather than its efficiency. (K Keen and Scott-Morton [1978]

Growths in computer and information technology and the realization of the inherent limitations of individual’s capability to take effective decisions in dynamic, unstructured and semi-structured decision situations, have naturally led to the development and use of various concepts and tools for using computer in making decisions.

Features of DSS:

1. Dss assists managers in their decision making specifically in semi – structured and unstructured fields.
2. It supports and enhances, rather than replaces, managerial decisions.
3. It combines the use of models and analytical techniques with conventional data access and retrieval functions.
4. It has enough flexibility to accommodate changes in the environment, the approach and the needs of the users.
5. It improves the effectiveness of the decisions rather than its efficiency.
6. It supports managers at all levels that take decisions.
7. Its user initiated and user controlled.
8. It support s the personal decision making style of individual managers.
9. It has features( including interactive features ) which makes its use by non computer people easy.

Q2. (a) Consider the employee database:

 Emp(empno,ename,job,sal,comm.,hiredate,deptno)

Dept(deptno,dname,loc)

Give an expression in SQL for the following:

1. Add a new record in emp table.

Ans. Insert into emp values(‘E01’,’xyz’,’manager’,40000,2013-10-10,10);

1. Increase 10 percent salary of employees working in deptno 10

Update emp set sal =sal\*1.10

Where deptno=10;

1. Display empno, job and dname for all the employees working in deptno 30

Select emp.empno,emp.job,dept.dname

From emp INNER JOIN dept

ON emp.deptno=dept.deptno;

1. Find the name of the employee having highest salary.

Select ename from emp where salary =(select max(salary) from emp);

(b) What is referential integrity constraint and how does it works?

Ans. Referential Integrity is set of constraints applied to foreign key which prevents entering a row in child table (where you have foreign key) for which you don't have any corresponding row in parent table i.e. entering NULL or invalid foreign keys. *Referential Integrity* prevents your table from having incorrect or incomplete relationship e.g. If you have two tables Order and Customer where Customer is parent table with [primary key](http://java67.blogspot.sg/2012/10/difference-between-primary-key-vs-candidate-keys-sql-database.html) customer\_id and Order is child table with foreign key customer\_id. Since as per business rules you can not have an Order without a Customer and this business rule can be implemented using *referential integrity in SQL* on relational database. **Referential Integrity** will cause failure on any INSERT or UPDATE SQL statement changing value of customer\_id in child table, If value of customer\_id is not present in Customer table.

 A FOREIGN KEY in one table points to a PRIMARY KEY in another table.

For example lets consider the following two tables:

The "Persons" table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_Id** | **LastName** | **FirstName** | **Address** | **City** |
| 1 | Hansen | Ola | Timoteivn 10 | Sandnes |
| 2 | Svendson | Tove | Borgvn 23 | Sandnes |
| 3 | Pettersen | Kari | Storgt 20 | Stavanger |

The "Orders" table:

|  |  |  |
| --- | --- | --- |
| **O\_Id** | **OrderNo** | **P\_Id** |
| 1 | 77895 |  3 |
| 2 | 44678 |  3 |
| 3 | 22456 |  2 |
| 4 | 24562 |  1 |

The "P\_Id" column in the "Orders" table points to the "P\_Id" column in the "Persons" table.

The "P\_Id" column in the "Persons" table is the PRIMARY KEY in the "Persons" table.

The "P\_Id" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

The FOREIGN KEY constraint also prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

Q3. a)Draw and explain the Data warehouse architecture for insurance sector.

Ans. A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources.

In addition to a relational database, a data warehouse environment includes an extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.

|  |
| --- |
|  |

\

 Data warehouse architecture

 Data warehouse architecture enables the ability to implement solutions that span many industry sectors, such as banking , insurance, retail,, health care ,telecommunications and government. It represents architecture for the enterprise and the capabilities to implement a resilient, adaptive architecture to enable and ensure high – performance and sustained value for the enterprise. The reference architecture provides a framework of components that can manage the life cycle of master data , manage the quality and integrity of the data, make master data actionable, and provides stateless services to control the consumption and distribution of data.



 Data warehouse architecture for insurance sector

Q3.(b) Why is partitioning of detailed data in data warehousing required?

Ans. Data warehouses often contain very large tables and require techniques both for managing these large tables and for providing good query performance across them. An important tool for achieving this, as well as enhancing data access and improving overall application performance is partitioning.

Partitioning offers support for very large tables and indexes by letting you decompose them into smaller and more manageable pieces called partitions. This support is especially important for applications that access tables and indexes with millions of rows and many gigabytes of data. Partitioned tables and indexes facilitate administrative operations by enabling these operations to work on subsets of data. For example, you can add a new partition, organize an existing partition, or drop a partition with minimal to zero interruption to a read-only application.

Partitioning can help you tune SQL statements to avoid unnecessary index and table scans (using partition pruning). It also enables you to improve the performance of massive join operations when large amounts of data (for example, several million rows) are joined together by using partition-wise joins. Finally, partitioning data greatly improves manageability of very large databases and dramatically reduces the time required for administrative tasks such as backup and restore.

Granularity in a partitioning scheme can be easily changed by splitting or merging partitions. Thus, if a table's data is skewed to fill some partitions more than others, the ones that contain more data can be split to achieve a more even distribution. Partitioning also enables you to swap partitions with a table. By being able to easily add, remove, or swap a large amount of data quickly, swapping can be used to keep a large amount of data that is being loaded inaccessible until loading is completed, or can be used as a way to stage data between different phases of use. Some examples are current day's transactions or online archives.

Q4.Diffference between:

1. OLTP and OLAP
2. Database and Knowledgebase
3. MIS and DSS
4. Neural Network and Genetic Algorithm

Ans. We can divide IT systems into transactional (OLTP) and analytical (OLAP). In general we can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.



- **OLTP (On-line Transaction Processing)** is characterized by a large number of short on-line transactions (INSERT, UPDATE, DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is the entity model (usually 3NF).

- **OLAP (On-line Analytical Processing)** is characterized by relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema).

The following table summarizes the major differences between OLTP and OLAP system design.

|  |  |  |
| --- | --- | --- |
|  | **OLTP System Online Transaction Processing (Operational System)** | **OLAP System Online Analytical Processing (Data Warehouse)** |
| Source of data | Operational data; OLTPs are the original source of the data. | Consolidation data; OLAP data comes from the various OLTP Databases |
| Purpose of data | To control and run fundamental business tasks | To help with planning, problem solving, and decision support |
| What the data | Reveals a snapshot of ongoing business processes | Multi-dimensional views of various kinds of business activities |
| Inserts and Updates | Short and fast inserts and updates initiated by end users | Periodic long-running batch jobs refresh the data |
| Queries | Relatively standardized and simple queries Returning relatively few records | Often complex queries involving aggregations |
| Processing Speed | Typically very fast | Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes |
| Space Requirements | Can be relatively small if historical data is archived | Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP |
| Database Design | Highly normalized with many tables | Typically de-normalized with fewer tables; use of star and/or snowflake schemas |
| Backup and Recovery | Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability | Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method |
|  |

b) Database and Knowledge Base

Q5.b)What are the applications of data mining in marketing and HRM?

Ans. Nowdays, the capability to collect data has been expanded enormously and provides enterprises huge amount of data. The interesting knowledge or the high valued information about the customer can be extracted by data mining. By following the market segmentation strategy, an enterprise could increase the unexpected profits.However, as for the enterprise the customer’s

http://ijcsn.org/IJCSN-2013/2-5/IJCSN-2013-2-5-47.pdf