Table of Contents

Task 1 [A]

Critically compare different data models and schemas. [P 1.1]

Introduction: Data Models

Data model describes the model of logical structure of database. According to Liberty University (n.d.) "*A data model is a conceptual demonstration of the data structures that are required by a database.*" It describes collection of at least three database components, data structure, general integrity and rules of inference that to be applied to instance of database.

Data model represents how data are connected to each other and rules (constraints) applied. It also defines how data will be managed and stored in the system. In simple word it is planning structure of database system. This report will critically compare different data models such as hierarchical, network and relations data models.

Hierarchical Data Model

Hierarchical data model is oldest data model between three data models. In this data model data is organized in family tree or inverted tree model. This data model contains two segments, child and parent. It creates Parent Child Relationships between records. Hierarchical supports 1: N (one-to-many) relation meaning parent can have multiple (N) child but child can have only one parent. This model provides database security, data independence and integrity but tends to be complex as well as has limitation due to inability of one to many relationships.
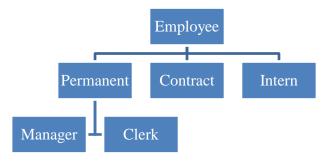
Figure 1 Representation of Hierarchical Data Model

Network Data Model

Like hierarchical model, network data model is also represented in family or inverted tree model. But the one of the key advantages of this type of data model is it is able to establish N: N (many to many) relationship. Network data model creates relationships called sets which consist of owner record type, member record type and set name. Member record type can have same role in different sets as well hence supports multi-parent concept which is its main advantage over first type of model. It is more flexible and supports more relations compared to hierarchical.
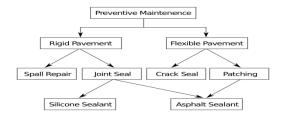
Figure 2 Representation of Network Data Model (Wikipedia.com)

Relational Data Model

Relation data model is latest data model compared to other two. Unlike other two models, this data model is represented in two dimensional relations called tables. Column represents attributes or domain and Row represents record. This data model allows maintaining integrity constraints, data structure and storage-retrieval. Similar to network model, it allows relation between different tables but with more rules (integrity). Since it reduces data redundancy and supports normalization modern software are using this kind of model.

Name	City	Contact
RAM	Kathmandu	9840002321
Shay	Pokhara	9843324236
Harry	Jabalpur	9844432230

Figure 3 Representation of Relation Data Model (Table)

Critical Comparison:

One of the key advantages of the first data model is that accessing records in database is fast due to the relation between child and parent is done through pointer. It is an efficient data model and adding and deleting records is really simple. But comparing with other data models it has its limitations such as, difficulties in data management, difficult implementation and programming complexity.

Network data model is more advanced as compared to hierarchical type. Easier data accessing and handling more relationship as well as data integrity is possible in this type of model. One of the key limitations of this model type over relation data model is lack of structural independence and procedural access language. It is also more complex data model.

Relation data model as compared with other data model is most advanced and efficient data model. Data consisting row and column makes data insert and editing attractive and easier. This model is flexible and more secured compare than both hierarchical or network type model.

One of the key advantage over other two model is data retrieval from several table is much easier just by joining them as desired. But like other models it has its own limitations such as it requires more powerful hardware than the other two. Another disadvantage can be since designing database with

this data model is much easier; sometimes it leads to less organized or poorly designed database system.

Schema and Instance

The term schema represents the overall structure of the database. Schema is design of database which describes the variable types, attributes in the table.

Instance in other hand is particular moment of information stored in that database. It represents value of variables unlike schemas which only represents variable type.

Critical comparison:

Both instance and schema represents different elements of database than each other. Instance resembles DML (data manipulation language) of database base such as insert, delete, update etc. of a certain moment. Instance is correspond values in cells of tables. On the other hand schema resembles DDL (data definition language) which describes definition of the database. Schema gives overview of how many attributes are there in table. Another key difference between schema and instance is schema of a database is less-modified (not ever) but same does not apply for instance.

Conceptual Schema and Physical Schema

Conceptual schema of database denotes structure of database in highly abstract nature. In this schema entities of database is drawn and attributes doesn't have much roles to play. Conceptual schema signifies how entities are related to each other. It is basic level of planning for database design. Conceptual schema provides first study of requirement of database.

Physical schema is more depth planning for database design. Here attributes for each entity are analyzed for implementation.
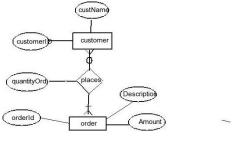
Figure 4 Conceptual Schema
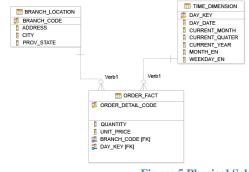
**Figure 5 Physical Schema
(www-01.ibm.com/)**

Critical comparison:

The basic similarity between both schemas is they both represent overview of database. And the key difference is the level of details they provide. To prepare physical schema, conceptual schema is prepared first. Conceptual schema is less-attributed and other one is fully-attributed.

Task 1 [B]

Critically discuss the benefits and limitations of different database technologies. [P1.2]

Introduction

In modern industries, there is large amount of data available. Medical data, financial data, historic data, market data, graphic data, governmental data etc. and all these data is useless until it's converted into useful information. Development in database technologies has helped to transform these large numbers of data into useful information.

Different database technologies manage these data in database differently. This makes managing and extracting data from database more efficiently that process becomes more quick and secured. In this report, benefits and limitations of different database technologies is discussed.

Data Warehousing

Data warehouse is computerized warehouse to store large number of information. In Data warehouse, information is taken from different business components and stores it in a way that enables analytical queries through a process known as Extract.
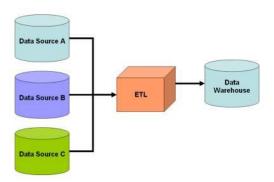
Figure 4 Data Warehouse (Source: chartio.com)

Benefits

- Data warehouse helps user to access large amount of optimized useful information
- Improvement in data quality and consistency.
- Reduce to cost of computing and increase revenue for company.
- Different types of data from different source can be stored in same centralized location
- Since data is efficiently managed, accessing and decision making so quicker which results time saving.

Limitations

Even though there is large number of benefits of data warehouse, it has its own limitations too. To store data in warehouse it need to go through ETL process as shown in figure 1 above. ELT

represents extract, transform and loading. This process can take time before data is to be stored. Another point need to be considered is if not handles properly and securely, access from outside in warehouse can cause various security issues. Data warehouse need maintenance and highly skilled user. Hence it can also be considered as one of few limitations of this technology.

Data Mining

Data mining is technology to retrieve required information from one or more data sources. It is also known as KDD –"knowledge discovery in database" and is process of analyzing data from various data activity patterns. It retrieve data from source based on past use pattern of data for example, if customer brought beer they also tend to buy coca cola. It is intelligent and automated method to retrieve summarized information for the company.
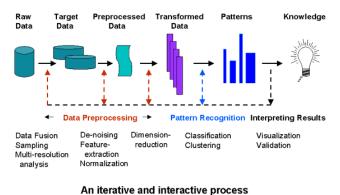
Figure 5 Data Mining Process (Source: computation.llnl.gov)

Benefits

There are some key benefits of data mining technology. Since it analyzes the pattern it can predict market status for marketers by providing them more accurate and useful prediction automatic and efficiently. It can help bankers to issue credit card or loan by analyzing the pattern of past clients behaviors, location, age, sex with the new applicant.

Another benefit of data mining is identifying suspects from analyzing habit, behaviors and other patterns. It can also help researchers by making data analyzing process faster.

Limitations

Like data warehouse, data mining technology has limitations too. Few limitations are listed here.

- Analyzing large pattern requires large information. This can cause privacy and security problem. Large number of personal information can be at risk if not handled securely.

- Larger and complex data processing and maintenance requires powerful hardware and software system.

- As it works on pattern recognition, it is not always hundred percent accurate.

Documents Management System

Document management system is computerized system to organize and store electronic documents such as text files or scanned images. Document management system often referred as DMS takes help of software to manage documents and provide user ability to access or modify data centrally. It is system that manages document of whole organization.
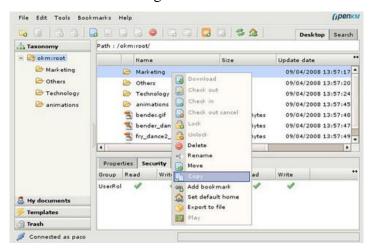
Figure 6 Open Source DMC (Source: www.web-delicious.com)

Benefits

One of the key benefits of document management system is centralized storage which enables user to access information from different regions of office. Other key benefits of DMS are it provides enhanced security features such as disaster recovery and access control. Since it is computerized and automated, less labor is required this cuts the cost of organization.

Limitations

Document management system also possesses some limitations in it. Some key limitations are listed below:

- Investment is required to set up the system.
- Poorly planned implementation can cause various problems.
- Staffs are required to train to understand new technology.

Web Enabled Database

Web enabled database is database system which is designed and developed to be accessed through internet. User are able to access into the database no matter where they are if they are connected though internet. Like other technologies it has its benefits and limitations which are listed below.

Benefits

- It saves the money for company as they need to pay only for as much of storage they use.
- It provides 24*7 access to database to user from anywhere with internet access

- It does not require special hardware/software system, a system with internet access and web browser can do all the task
- Auto data backup

Limitations

- Uploading data into web means it requires trust; security is big concern in this technology.
- Requires internet connection, if for some reason connection breaks, no work can be done on database.
- Need to pay subscription fee in monthly, annually basis to service provider.

Multimedia Database

Multimedia data represents video, image, audio, animation or graphics data. Multimedia database is required to manage increase number of these data in modern days. It is also represented as MMDS.

Benefits and limitations

One of the key benefits of multimedia database is it helps to organize large number of multimedia data. It is easy to use as it is user friendly, flexible and can be used for many audiences.

On the other hand it has some limitations such as it is expensive than other database technologies. Multimedia data takes time to process hence this technology is more time consuming.

Summary

Above in this report, benefits and limitations of several database technologies such as web enabled, data mining, data warehouse etc. were critically discussed. It is found that every technology has its own benefits and limitations. Company should analyze requirements, cost before choosing the database technology for their business. These technologies can enhance their business process and increase revenues along with saving time and cost.

If company takes care of security, maintenance and training of their staff, no matter what database technology they use, it will help them to grow their business.

Task 1[C]

Analyze different approach to database design for a given scenario. [P1.3, M1]

Database Normalization

Studytonigh (n.d.) states normalization is systematic approach to decompose a data tables to eliminate data redundancy. Normalization ensures data stored in tables are useful and store structure is logically correct. It is step by step process and in each step level different kind of data redundancy and anomalies are eliminated.

Some of the benefits of normalization listed by Maan (2012) in his personal blog are listed here:

1. Elimination of duplicated data from database results smaller and well-structured database. This result effective database architecture which consumes less storage.

2. Smaller tables mean fewer columns (attributes) in each table. Hence query for insert, select, delete, update are smaller which is less prone to error.

3. Maintaining database is easier and quicker

Normalization up to 3NF

1NF

1NF is first stage of normalization. It sets the fundamental rules for the database normalization. The first normal form (1NF) states:

- Each attributes in a table must be unique
- No row or column can be duplicate
- Each table must have primary key
- No row or column can have multivalued data

2NF

Second normal form is second phase of database normalization. It adds levels of normalization to database that are in 1NF state. 2NF ensures these facts:

- Only related data are in a table.
- No data redundancy is possible anymore.

Database system for the given scenario is normalized up to 3NF. 3NF is level of normalization done. According to Microsoft (2013) 3NF is process of elimination of columns which are not depend on the key. Here are some key requirements for table to be in 3NF normalization:

- Table should be in 2NF
- Transitive functional dependency should be removed.
- All other attributes in table should depend on the primary key.

For example, the developed design for database has table tblStudent 3NF normalized. Here are 3NF normalized tables for the database. Every table only has attributes that are dependent on primary key of that table.

1. tblStudent { StudentID, guardianID, Enroll_Type, Name, Address, Gender, Email, Contact, Photo}
2. tblGuardian{GuardianID, Name, Address, Contact}
3. tblDepartment{DepartmentID, Name, HOD}
4. tblGrade {StudentID, UnitID, Mark}
5. tblStudying {StudentID, ProgramID, Semester}
6. tblPrograms{ProgramsID,Name, Duration, CreditHour}
7. tblEnroll_Type {Enroll, Description}
8. tblUnit{unitID,Name, CreadiHour,TeacherID, ProgramID}
9. tblTeacher{TeacherID,Name,Address,Gender, Email, Contact, Salary, DepartmentID, Designation}
10. tblSemester {Semester, Description}

E-R Diagram

E-R diagram is a graphical representation of entities and their relation in database Webopedia (Beal, n.d.). It has symbols that represents entities, relations and attributes, ER diagram describes how each of them are connected to each other. Thakur (n.d.) notes some of the key benefits of E-R diagram.

- Drawing of ER diagram for the system helps the database development much easier. Conversion of E-R diagram to relation diagram is very simple.

- Conversion of E-R diagram to other models such as hierarchical or network model is much easier.

- Though it is not industry standard diagram it is very popular in high level design.
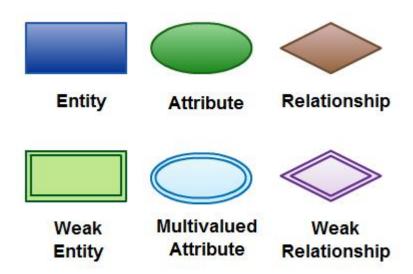
Figure 7 Symbols in ER diagram (source: http://creately.com/blog/wp-content/uploads/2012/03/ER-Diagram-Elements.jpeg)

Data Flow Diagram

Data flow diagram (DFD) as name suggest is a graphical representation of flow of data in the system. According to Sybase (2012) Data flow diagrams are used by systems analysts to design information-processing systems. It represents how system would works in different level. Key elements in DFD are process, flow, store, and terminator.

Some of advantage of data flow diagram is listed here below.

- Detailed Processes
- Precision of Flow
- Suitable Data Store
- Levels of Data Flow Diagram

Symbols used in Data flow diagram and DFD for the database system to be developed is drawn in separate pages below.

Element	Meaning
Data Store	Data stores or places where data is held temporarily or permanently.
External Entity	External entities such as people or other systems that feed data to or take data from the system. If the external entity is a system, remove the person from the box.
Process A	A process that manipulates data.
Data	The flow of data between data stores, external entities, and processes.

Figure 8 Symbols used in Data Flow Diagram (DFD)
Source: http://requirements.seilevel.com/blog/wp-content/uploads/2014/07/DFD-Symbols.png

Schema Diagram

Janssen (n.d.) defines schema diagram as visual representation of tables, keys, relation etc. Table relationship is described via line joining primary key from parent table to foreign key from child table. It provides overview of the database.

ER-Diagram, Data Flow diagram and Schema diagram for the system is drawn in separated pages below.

References

- Beal, V. (n.d.) Entity Relation Diagram Available: http://www.webopedia.com/TERM/E/entity_relationship_diagram.html Accessed [10/25/2014]

- Cory Janssen

- Janssen, C. (n.d.) *Schema* Available: http://www.techopedia.com/definition/1242/schema Accessed [10/25/2014]

- Maan, D. (2012) *Advantage-disadvantage of normalization* Available: http://sqlserverwithbi.blogspot.com/2012/10/advantages-disadvantages-of.html Access [10/25/2014]

- Microsoft (2013) *Description of the database normalization basics* Available: http://support.microsoft.com/kb/283878 Accessed [10/25/2014]

- Studytonigh (n.d.) *Database normalization* Available: http://www.studytonight.com/dbms/database-normalization.php Accessed [10/25/2014]

- Sybase (2012) *Data flow Diagram* Available: http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc38088.1610/doc/html/rad1232026266129.html Accessed [10/25/2014]

- Thakur, D. (n.d.) Advantage and disadvantage of E-R model Available: http://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-e-r-model Accessed [10/24/2014]

Task 2 [A]

Design a relation database system to meet a given requirement in the scenario and apply a range of database tools and techniques to enhance the user interface. [P2.1, P2.2, M2]

Introduction: Requirement documentation

The purpose of this document is to develop requirement specification of the database system to be developed for ISMT College. Requirement document provides overview of outcome of system development and its requirements and without providing technical details. In simpler words, requirement document explains **what** system should do rather *how it should do*. Requirement documents are that document that explains requirements to develop a system such as the purpose of system to be developed, its context, costs, future requirement, time scales and how the system will look like when development process ends.

Requirements are those properties, constraints or functions that must be met or satisfied by system to fulfill the need of intended system's user, Wright (2008). It is list of tasks that need to be done to get desired solution.

Deep analyses of requirements are needed to be carried out before developing a system. To find requirements of system, several finding methods are available. Below, this report will discuss few useful fact finding tools and apply them to discover requirement of the system to be developed and then prepare Requirement Documentation.

Fact Finding Methods

Fact finding is process of gathering required information for specific purpose. While gathering requirements, these come from different stakeholder such as users, customers or even system developer. Hence interaction with them to discover detailed information regarding requirement is necessary. Though there are also fact finding methods that may not require direct interaction with stakeholders.

Interview

Basically interview is planned meeting with individual or group to collect required information, Shah (2008). It is most common method in which, analyst asks range of different question with individual to obtain information needed. Several things are considered before planning interview session such as:

- People to interview

It is important to determine whom to interview. It is necessary to pick right people to obtain right information for example, for vehicle management system, security guard can be interviewed but same in is not required while developing academic system.

- Objectives for the interview

After people to be interviewed are decided, overview for the outcome of interview should be planned out. It is the phase when what kinds of information need to be collected from result of interview are planned out.

- Interview questions.

In this phase, questions for the interview are prepared. Different questions are prepared for different individual/group for efficient outcome of interview.

Questionnaires

Questionnaires are another type of fact finding method in which set of questions is distributed among stake holders. Information is analyzed based on the response in questionnaire. It is a short and specific way to find the information. Some of the benefits of it are listed here below.

- Large information can be gained from stake holders in small amount of time

- Cost effective

- Helps to get views on the topic from different people before making changes or developing a system

Record View

Record view is very effective technique to gather useful information. In this type of fact finding method, different records such as attendance sheet, payment receipt, user logs, history data are analyzed.

Observation and document checking

In this technique information is taken from observing the area, document, website, work environment etc. Result is achieved via observation of different elements in the system such as document flow, how payment is done, how data is kept in past etc.

Applied Methods

For requirement specification of the database system to be developed, different types of fact finding tools and techniques are deployed. These tools, how they are deployed and result of the analysis are described below.

Interview

To find the requirement for the system to be developed, different stake holders of the college were interviewed. Questions asked in the interview are listed below.

1. How data is collected in current system?

2. Why current system is not competent enough?

3. Do you have knowledge of database system?

---

4. Does college have man power to manage system developed using SQL server database?

5. Is college ready to invest on database management system? If yes how much can they invest?

6. How much time college can give to develop the system?

7. Does college want to store financial record on database?

8. Does College want to store academic data on database?

9. Will college need regular support on database maintenance?

10. Does college have other system that can be integrated with database system to be developed?

<u>Document checking</u>

College Website:

Other documents observed Payment Receipts, Prospectues, Attendance Sheet are attached below.

Observe Sheet

S.N.	Observe object	What was observed	Observe result
1.	Attendance Sheet	How student are managed in class	Each student has unique id
2.	Prospectus	How student are enrolled	College different Programs, students are enrolled in different program
3.	Website	Grading system	Each student is graded based on their unit assignment result.
4.	Receipt	How academic, admissions, exam time table are managed.	College is semester system based.

Questionnaires

Some questions provided in the questionnaires paper is listed below.

1. Is current document keeping system is satisfying.

   A. Yes

   B. No

2. Is current system able to provide quick service?

   A. Yes

   B. No

3. Is current safe and secure?

   A. Yes

   B. No

   C. Can be safe if handled properly

4. Has data lost due to incompetence of the current system?

   A. Yes, a lot

   B. No, a lot

   C. Occasional

5. Is database system is really required?

   A. Yes

   B. No

6. What operating system is college using currently that can support database system?

A. Microsoft Windows

B. LINUX/UNIX

C. MAC

7. How much RAM the current system has

A. 4 GB

B. 8 GB

C. 16 GB

D. Other

Documentation

Documentation of the requirement specification is based on the result found from the tools and techniques as listed above in this document.

Relation Requirement (Entity required in database)

S.N.	Entity	Purpose
1.	tblStudent	To Store student's information
2.	tblGuardian	To store Guardian's information
3.	tblDepartment	To store department's information
4.	tblGrade	To Store Grade information
5.	tblStudying	To store student's status information
6.	tblPrograms	To store Program's information
7.	tblEnroll_Type	To store enrollment type
8.	tblUnits	To store units information
9.	tblTeacher	To store teachers information
10.	tblSemester	To store semester information

Technical Requirement

Hardware Speciation (Recommended)

1. RAM   - 4GB

2. Processor   - Intel I3

3. Hard Disk   - 320 GB

Software Specification

1. Operating System   - Windows 7 or later
2. Applications-
      a. Microsoft SQL server 2008 or later (For database system)
      b. Visual Studio 10 or later (For UI)

Cost

To estimate the cost of the system project is broken down in to smaller phases. Cost for each phases are calculated. Breaking down the project also helps to ease the development process.

S.N.	Task	Work Force Rate	Total Hour Required	Sub Total
1.	Analysis	7$ /HR	10 HR	70$
2.	Planning	15$ HR	15 HR	225$
3.	Design	10$ HR	10 HR	100$
4.	Implementation	8$ HR	50 HR	400$
5.	Maintenance	150$/YR	-	100$/YR
6.	Training	100$	-	100$
			Total	895$+150$/YR
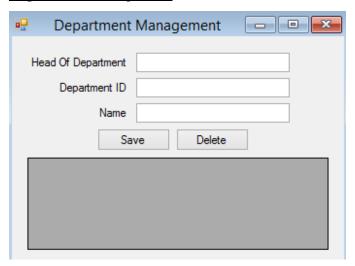
Future Requirement

1. Hard disk: If space in current hard disk fills up new hard disk is need to be attached to the system if more data is to be stored in database.
2. Training: User of the system in college needs training in time-to-time basis. If system is upgraded later, user must get trained first before using the system.
3. Annual Maintenance
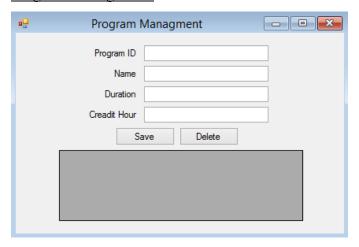
User Interface

User interface is designed using visual studio windows form.
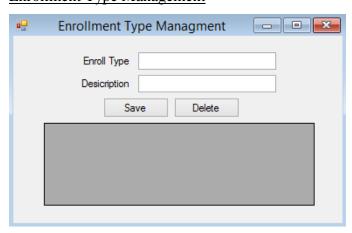
Department Management
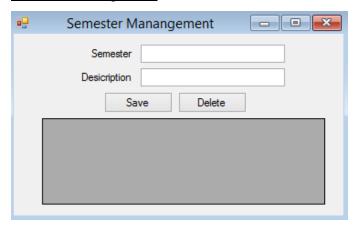
Program Management

Enrollment Type Management
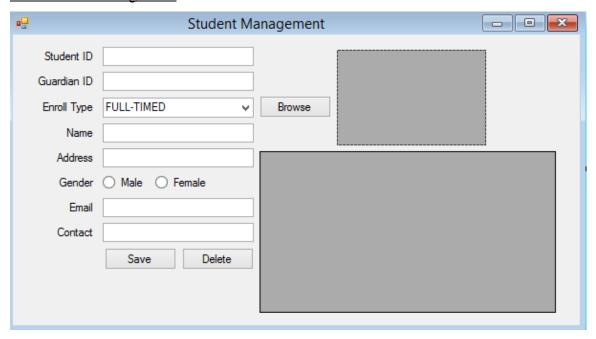
Semester Management
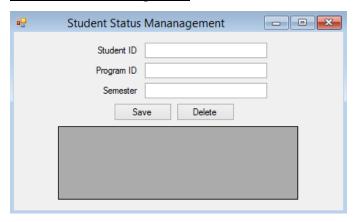
Guardian info Management
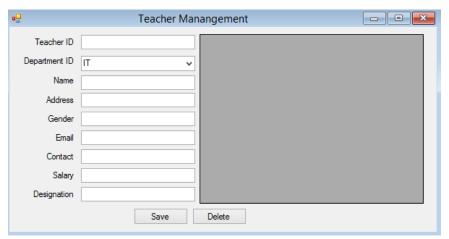
Student info Management
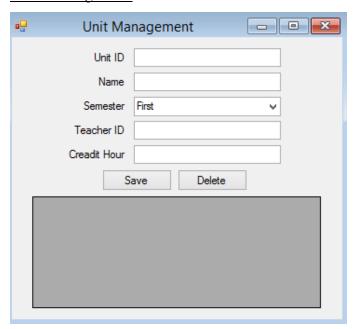
Student Status Management

Student Status Mananagement

Student ID

Program ID

Semester

Save   Delete
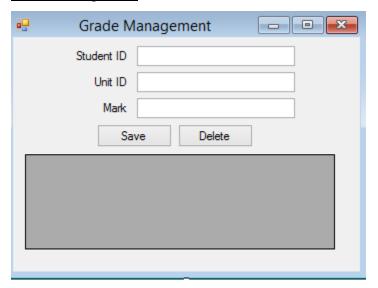
Teacher info Management

Teacher Manangement

Teacher ID

Department ID   IT

Name

Address

Gender

Email

Contact

Salary

Designation

Save   Delete

Units Management

Unit Management

Unit ID

Name

Semester   First

Teacher ID

Creadit Hour

Save   Delete

Grade Management

Timescale

Timescale for the development phase of database system is displayed below in another page with help of Gantt chart.

Reference

- Shah, B. (2008) *Fact Finding Techniques* Available:
  http://systemanalysisanddesign.blogspot.com/2008/11/fact-finding-techniques.html Accessed
  [9/10/2014]

- Wright, D. (2008) *Cascade* Published: Lulu.com

Task 2 [B]

Build relational database system based on the prepared design. [P2.2, D2]

Introduction

Structured Query Language also known as SQL is type of programming language that is used for managing databases. SQL is consists of DDL and DML. Data definition language or DDL handles the structure of a database. It is basically used for creating, modifying and destroying a database and objects in it. The main commands in DDL are:

Create

This is used for create new database or objects in database.

Alter

This command is used for updating (modifying database elements)

Use

Use is used for commanding the system to use certain database

Drop

Drop is used for either dropping (deleting) database or objects from it.

SQL Server Management Studio

SQL server management studio is a tool built for windows Computers that provides platform to manage databases. This Microsoft product allows performing tasks such as creating, deleting, altering databases, tables and data.

Some of the tools for managing databases are oracle, MySQL and MS Access. For the current system to be developed, Microsoft SQL server is used due to availability and rich features.

Data Type

Data type is something that describes what kind of data is that can hold. For example data type can be an integer, character, binary or date type.

Some most used data types in database developing are listed below in table.

S.N.	Data Type	Description
1	Int	Can hold -2147483648 to 2147483647
2	Char	Can hold 1 character
3	Nvarchar(size)	Can hold string no given size
4	Date	Can hold date
5	Image	Can hold binary data of up to 2 GB

Entity and Attributes

In database system, entity is object that can relate to real world such as student, teacher, program etc.

Meanwhile attributes are that property which describes entity. Such as name of student, his/her roll

number, age etc.

Keys and Identifier

Keys are individual or set of attributes that ensure any record within a table can be uniquely identified. There are different types of keys such as super key, candidate key, primary key, foreign key etc.

Identifier is basically unique name of database, tables and attributes. Name of views, constraint,

triggers etc. are also defined as identifiers.

Domain and Referential Constraints

A domain constraint defines rules in relational table that allows certain type of value in attributes.

For example, a domain of char only allows 1 character, a domain of binary only accepts binary value,

integer domain only accept whole number.

Similarly referential constraints are that rule that defines rules in relationship between two or more

entity.

Building Relational Database for prepared design

Entity and Attributes Considered according to ER-Diagram:

S.N.	Entity	Attributes	Not Null	Data Type	Key
	tblStudent	guardianID	Yes	Int	Foreign
		Enroll_Type	Yes	nvarchar	Foreign
		StudentID	Yes	Int	Primary
		Name	Yes	Nvarchar	-
		Address	-	Nvarchar	-
		Gender	-	Nvarchar	-
		Photo	-	Nvarchar	-
		Email	-	Nvarchar	-
		Contact	-	Nvarchar	-
		Photo	-	Image	-
2	tblGuardian	GuardianID,	Yes	int	Primary

		Name,	Yes		Nvarchar	-
		Address,	-		Nvarchar	-
		Contact	-		Nvarchar	-
3	tblDepartment	DepartmentID	Yes		Int	Primary
		Name	Yes		Nvarchar	-
		HOD	Yes		Nvarchar	-
4	tblGrade	StudentID	Yes		Int	Foreign
		UnitID	Yes		Int	Foreign
		Mark	-		decimal	-
5	tblStudying	StudentID	Yes		Int	Foreign
		ProgramID	Yes		Int	Foreign
		Semester	YEs		Int	Foreign
6	tblPrograms	ProgramsID	Yes		Int	Primary
		Name	Yes		Nvarchar	-
		Duration	-		Int	-
		CreaditHour	-		Int	-
7	tblEnroll_Type	Enroll	Yes		nvarchar	Primary
		Description	-		nvarchar	-
8	tblUnits	UnitID	Yes		Int	Primary
		Name	Yes		Nvarchar	-
		CreaditHour	-		Int	-
		TeacherID	Yes		Int	Foreign
		ProgramID	Yes		int	Foreign
9	tblTeacher	TeacherID	Yes		int	Primary
		Name	Yes		Nvarchar	-
		Address	-		Nvarchar	-
		Gender	-		Nvarchar	-
		Email	-		Nvarchar	-

		Contact	-		Nvarchar	-
		Salary	-		Decimal	-
		DepartmentID	Yes		Int	Foreign
		Designation	-		nvarchar	-
10	tblSemester	Semester	Yes		Nvarchar	Primary
		Description	-		nvarchar	-

Data Type

```
(
UnitID int not null
Name nv
CreditH   data type int
TeacherID int not n
```

As the image above shows, data type in DDL is followed by attribute name. Each attribute is given appropriate data type here to achieve the requirement.

Keys and Identifier

Keys

```
constraint PK_StudentID Primary Key (StudentID),
constraint FK_GuardiaID Foreign Key (GuardianID) references tblGuardian(GuardianID),
```

Identifier

Domain and Referential Constraints

```
amID int not null,
nvarchar(30) not null,
ion int
```

Domain Constraint
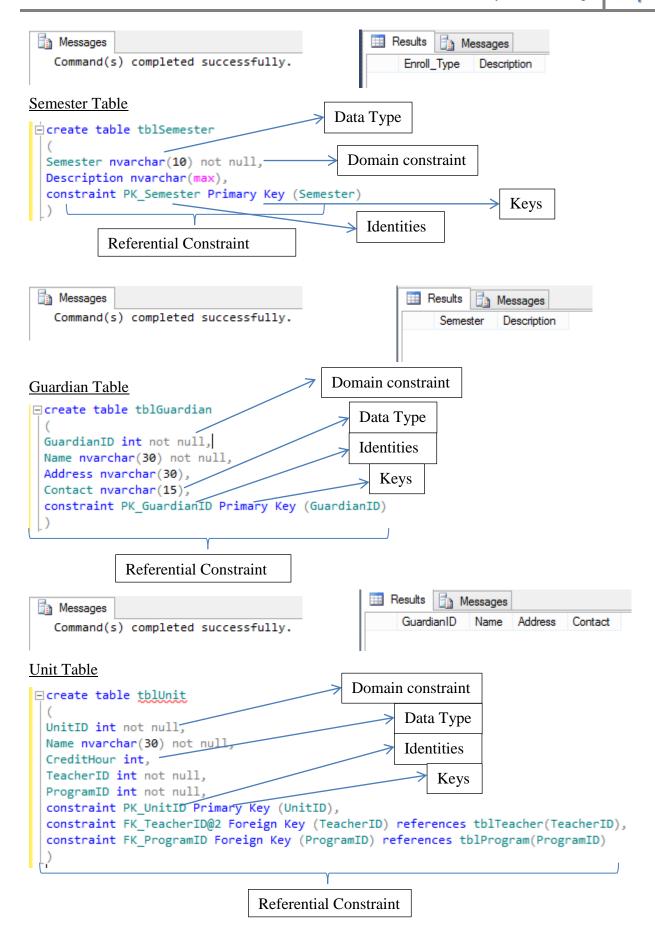
```
constraint FK_GuardiaID Foreign Key (GuardianID) references tblGuardian(GuardianID),
```
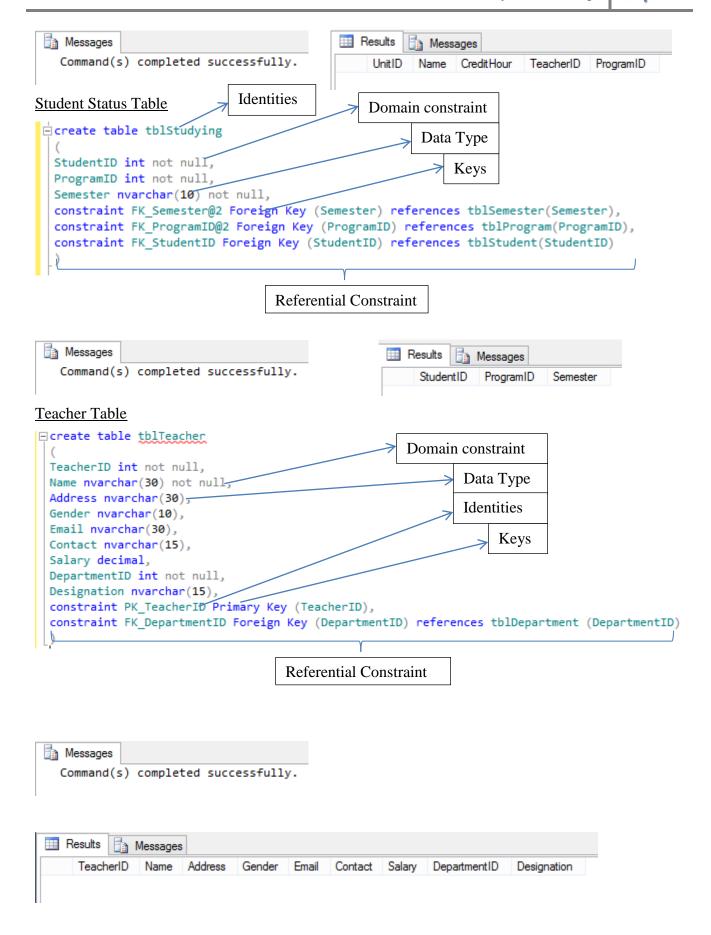
Referential Constraint
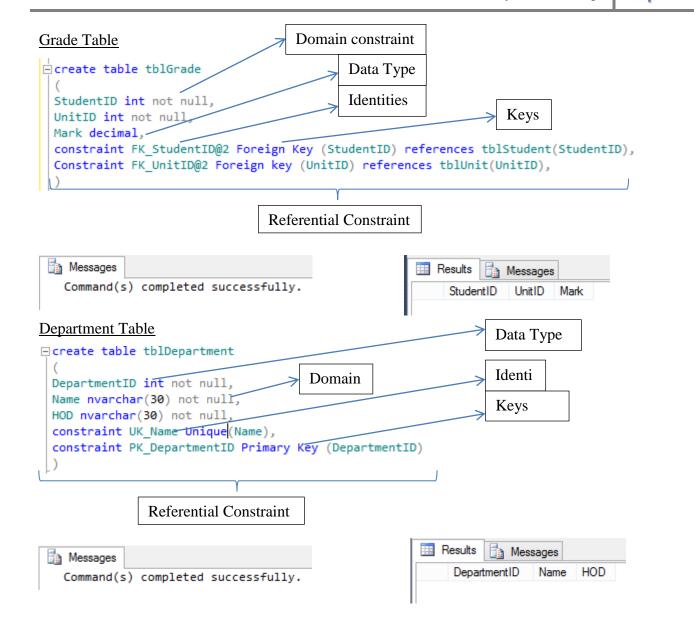
Entity and Attributes according to ER-Diagram

Student table

```
create table tblStudent
(
StudentID int not null,
GuardianID int not null,
Enroll_Type nvarchar(10) not null,         → Domain constraint
Name nvarchar(50) not null,
Address nvarchar(30),                        → Data Type
Gender nvarchar(10),
Email nvarchar(MAX),
Contact nvarchar(15),                        Identities
Photo image,                                              Keys
constraint PK_StudentID Primary Key (StudentID),
constraint FK_GuardiaID Foreign Key (GuardianID) references tblGuardian(GuardianID),
constraint FK_EnrollType Foreign Key (Enroll_Type) references tblEnroll_Type(Enroll_Type),
)
```

Referential Constraint

Messages
Command(s) completed successfully.

Results	Messages							
StudentID	GuardianID	Enroll_Type	Name	Address	Gender	Email	Contact	Photo

Program Table

Identities

```
create table tblProgram
(
ProgramID int not null,            → Domain constraint
Name nvarchar(30) not null,
Duration int,                       → Data Type                Keys
CreditHour int,
constraint PK_ProgramID Primary Key (ProgramID),
)
```

Referential Constraint

Messages
Command(s) completed successfully.

Results	Messages		
ProgramID	Name	Duration	CreditHour

Enrollment Type Table

Data Type

```
create table tblEnroll_Type
(
Enroll_Type nvarchar(10) not null,       → Domain constraint
Description nvarchar(max),                         Keys
constraint PK_EnrollType Primary Key (Enroll_Type)
)
```

Identities   Referential Constraint

```
Messages
   Command(s) completed successfully.
```

```
Results | Messages
   Enroll_Type | Description
```

Semester Table

```
create table tblSemester
(
Semester nvarchar(10) not null,
Description nvarchar(max),
constraint PK_Semester Primary Key (Semester)
)
```

Data Type

Domain constraint

Keys

Identities

Referential Constraint

```
Messages
   Command(s) completed successfully.
```

```
Results | Messages
   Semester | Description
```

Guardian Table

```
create table tblGuardian
(
GuardianID int not null,
Name nvarchar(30) not null,
Address nvarchar(30),
Contact nvarchar(15),
constraint PK_GuardianID Primary Key (GuardianID)
)
```

Domain constraint

Data Type

Identities

Keys

Referential Constraint

```
Messages
   Command(s) completed successfully.
```

```
Results | Messages
   GuardianID | Name | Address | Contact
```

Unit Table

```
create table tblUnit
(
UnitID int not null,
Name nvarchar(30) not null,
CreditHour int,
TeacherID int not null,
ProgramID int not null,
constraint PK_UnitID Primary Key (UnitID),
constraint FK_TeacherID@2 Foreign Key (TeacherID) references tblTeacher(TeacherID),
constraint FK_ProgramID Foreign Key (ProgramID) references tblProgram(ProgramID)
)
```

Domain constraint

Data Type

Identities

Keys

Referential Constraint

Messages
Command(s) completed successfully.

Results **Messages**

UnitID	Name	CreditHour	TeacherID	ProgramID

<u>Student Status Table</u>

Identities

Domain constraint

Data Type

Keys

```
create table tblStudying
(
StudentID int not null,
ProgramID int not null,
Semester nvarchar(10) not null,
constraint FK_Semester@2 Foreign Key (Semester) references tblSemester(Semester),
constraint FK_ProgramID@2 Foreign Key (ProgramID) references tblProgram(ProgramID),
constraint FK_StudentID Foreign Key (StudentID) references tblStudent(StudentID)
)
```

Referential Constraint

Messages
Command(s) completed successfully.

Results **Messages**

StudentID	ProgramID	Semester

<u>Teacher Table</u>

```
create table tblTeacher
(
TeacherID int not null,
Name nvarchar(30) not null,
Address nvarchar(30),
Gender nvarchar(10),
Email nvarchar(30),
Contact nvarchar(15),
Salary decimal,
DepartmentID int not null,
Designation nvarchar(15),
constraint PK_TeacherID Primary Key (TeacherID),
constraint FK_DepartmentID Foreign Key (DepartmentID) references tblDepartment (DepartmentID)
)
```

Domain constraint

Data Type

Identities

Keys

Referential Constraint

Messages
Command(s) completed successfully.

Results **Messages**

TeacherID	Name	Address	Gender	Email	Contact	Salary	DepartmentID	Designation

Grade Table

Domain constraint

Data Type

Identities

Keys

```
create table tblGrade
(
StudentID int not null,
UnitID int not null,
Mark decimal,
constraint FK_StudentID@2 Foreign Key (StudentID) references tblStudent(StudentID),
Constraint FK_UnitID@2 Foreign key (UnitID) references tblUnit(UnitID),
)
```

Referential Constraint

Messages
Command(s) completed successfully.

Results | Messages
StudentID | UnitID | Mark

Department Table

Data Type

Domain

Identi

Keys

```
create table tblDepartment
(
DepartmentID int not null,
Name nvarchar(30) not null,
HOD nvarchar(30) not null,
constraint UK_Name Unique(Name),
constraint PK_DepartmentID Primary Key (DepartmentID)
)
```

Referential Constraint

Messages
Command(s) completed successfully.

Results | Messages
DepartmentID | Name | HOD

Task 3 [A]

Explain the benefits of using manipulation and query tools in a relational database system. [P3.1]

Introduction to DML

Like Data definition language (DDL), Data manipulation language also termed as DML is another type of category of SQL language. It is that part of SQL that allows manipulating data in database and database objected created with DDL. This manipulation includes inserting, selecting, deleting or updating etc. Below here in this document, different DML query commands and their benefits are explained.

According to WISEGEEK (2014), the core benefit of DML is that it allows user to manipulate data with database and without it data manipulation is not possible. It is the standard language to access the data in database. If data stored in database cannot be manipulated, that database becomes useless.

Here are some key benefits of data manipulation language:

1. DML commands allow modifying the data in a database.
2. User can find and retrieve the required data
3. Without DML other software cannot integrate with database system.
4. Allow deleting unwanted data

DML Query Commands

SELECT

Select is most common command and used for retrieving specific data from database. If there is data that matches the condition it returns all data else returns blank table.

Example:

*SELECT * FROM Persons*

Here in above example, select command list all information of persons from person table as there is no condition to filter out result.

SELECT Name, City FROM Persons where age =20

Now, in this example query returns only name and city of persons who is of age 20 because other persons are filtered by condition.

Some benefits of Select command:

o Let's users to retrieve data from database.
o Finds data that matches the condition
o Without select query command displaying data from database in any kind of application would not be possible.

o Let's user to filter out unwanted data with help of clause like 'where', 'and' , 'or',

INSERT

Insert query allows inserting new data row in table.

Example:

INSERT INTO Persons (PID, Name, City, Contact) VALUES (1, 'Jones', 'Katmandu', 9831333123)

Query in example above up on executing inserts a new row in table Persons and fills with given data in query.

Benefits of Insert Query command:

o Allow inserting new row in table as long as that does not break table's rule.

o Allows user to insert data on only required column leaving other as 'null'

o Without insert query user would not able to insert new data in database.

UPDATE

Update is command used for updating (editing/modifying) row(s) in data table.

Example:

UPDATE Persons set City= 'LONDON' where PID=1

Above query is for updating row of table Person and set city to LONDON and where clause allows update in only row where PID is 1.

Benefits of Update query:

o User can change/modify already inserted data.

o User can filter out row to modify with help of Where clause.

o User do not need to delete and insert new row just to modify a data in row

o Can modify data from bulk of rows at once.

DELETE

Delete command is used for deleting row(s) from table as requirement.

Example:

Delete From Persons Where PID=1

Query above deletes row(s) from Person table among rows which has PID 1 as filtered by Where clause.

Benefits of Delete Query:

o Allows to delete row from data table

o Allow deleting single of multiple rows that matches the condition(s)

o Without delete query unwanted data from data table cannot be removed.

JOIN

Join is SQL query that allows retrieving that from two or more than two table at once based on the relationship between columns in those tables.

Example:

SELECT Persons. Name, Persons. City, Orders.OrderNumber

FROM Persons

INNER JOIN Orders

ON Persons.PID=Orders.PID

Above query returns list of Person's name and city from Persons table as well Order number from Orders table based on rows that same PID in both tables.

Benefits of JOIN:

- o Allows to find data based on relationship between two or more tables
- o Allows to filter out unwanted rows with help of Where clause.
- o A single query can find complex data hence user do not need to use multiple select queries which helps software development

Aggregate function

These are functions in SQL that calculate column's data in table and return single value.

There are seven basic aggregate functions and they are:

- o AVG() - Gives the average value
- o COUNT() - Gives the number of rows
- o FIRST() - Gives the first value
- o LAST() - Gives the last value
- o MAX() - Gives the largest value
- o MIN() - Gives the smallest value
- o SUM() - Gives the sum

Example:

SELECT AVG(AGE) AS AGEAverage FROM Persons

Returns: Average of data in AGE column as AGEAVerage.

SELECT Count(PID) AS PIDCount FROM Persons

Returns: Number of columns that has PID and place them in column name PIDCount.

SELECT First(Name) AS FirstName FROM Persons

Returns: Name in First row of Persons Table and place them in column name FirstName.

SELECT LAST(Name) AS LastName FROM Persons

Returns: Name in Last row of Persons Table and place them in column name LastName.

SELECT MAX(AGE) AS AGEMAx FROM Persons

Returns: Max age of Person and place them in column name AGEMax

SELECT MIN(AGE) AS AGEMin FROM Persons

Returns: Min age of Person and place them in column name AGEMin

SELECT SUM(Salary) AS SalarySum FROM Persons

Returns: Sum of salary in all columns of Personals table and place them in column name SalarySum

Benefits of Aggregate funtions:

- o Allows to find minimum/maximum/sum/average/first/last and number of columns in table
- o Helps in data manipulation by easing out filtering unwanted and retrieving wanted row.
- o Single query can find minimum/maximum/sum/average/first/last hence extra coding is not required making programming simple.

References

Wisegeek (2014) *what is data manipulation language* Available: http://www.wisegeek.com/what-is-data-manipulation-language.htm Accessed [9/25/2014]

Task 3 [B]

Implement a query language into the relational database system. [P3.2, D3]

List all the part-time students enrolled in the BSc IT course

Self-Evaluation:

To solve the above problem, Select DML query is needed to be used for retrieving list of students who is part time and joined in BSc IT. Since information about program name is stored in Program table similarly information about student is stored in student table. But student and program are relational in student status table. Hence, data is required to be retrieved from three different tables forming i.e. form join of three tables. This query needs where clause to filter only student form program name BSc IT and enrollment type of part-time.

First, rows in student table are filtered by checking studentID in both student and student status table. After that, rows in program table need to be filtered out by comparing ProgramID in of both student status and Program table. Now Program of name BSc IT and student enrollment type of part-time should be filtered to get the required solution.

Solution:

Select tblStudent. Name, tblStudent.Address from tblStudent, tblStudying, tblProgram where tblStudent. StudentID=tblStudying.StudentID and tblStudying. ProgramID=tblProgram.ProgramID and tblProgram. Name='Bsc IT' and tblStudent. Enroll_Type='Part-Time'

Evaluation:

Result

```
(26 row(s) affected)
```

	Name	Address
1	Ajit	Kathmandu
2	Lane	Sooke
3	Madaline	Metairie
4	Tallulah	Geraldton-Greenough
5	Aubrey	Maasmechelen
6	Kirsten	Houtain-le-Val
7	Lavinia	Gonnosnò
8	Lucian	Nadrin

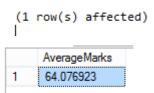**Calculate the average mark of the BHM students**

Self-Evaluation

To calculate average value of a column AVG() aggregate function are used. Since problem asks to calculate the mark of student avg(Mark) should be done. But since marks, program name are in different tables, join is required to solve the problem. First, rows in grade table are filtered with student that are also in student status table. After that, rows in program table need to be filtered out by checking ProgramID in of both student status and Program table. Now only program named BHM are filtered.

Solution

Select avg(G.Mark) as AverageMark from tblGrade G,tblStudying X,tblProgram P where G.studentID=X.StudentID and P.ProgramID=X.ProgramID and P.Name='BHM'

Result

```
(1 row(s) affected)
```

	AverageMarks
1	64.076923

Calculate the number of student who joined BTTM

Self-Evaluation

To count the number of student joined BTTM numbers of rows after filtering BTTM student are required to be counted. To do so Count () aggregate function should be used. Join is formed with tables program (to filter program name BTTM) and student status table (to establish relation between student and program).

solution

SELECT COUNT(S.StudentID) AS BTTMStudents FROM tblStudying S, tblProgram P WHERE P.ProgramID=S.ProgramID and P.Name='BTTM'

Result

```
(1 row(s) affected)
```

	BTTMStudents
1	39

Calculate the maximum mark gained by full-time student of BBA

Self-Evaluation

To calculate maximum mark gained by full time student from BBA max of mark after filtering BBA

and full-time student are required to be calculated. To do so, MAX () aggregate function should be used on mark column.

Program named BBA from Program table and Full-Time student from student table are required to filtered. For that data is required to be filtered from four different tables. Grade, program, student and student status.

Solution

select MAX(G.Mark) as MaxBBA from tblStudying ST, tblGrade G, tblProgram P, tblStudent S where ST.ProgramID=P.ProgramID and ST.StudentID=G.StudentID and P.Name='BBA' and ST.StudentID=S.StudentID and S.Enroll_Type='Full-Time'

Result

```
(1 row(s) affected)
```

	MAXBBA
1	99

Create View of all the student who has achieved more than 40 grade

Self-Evaluation

To create view of student who got marks above 40, a view named as student above forty marks should be created from grade table. Students must be filtered using Where clause and > operator.

Solution
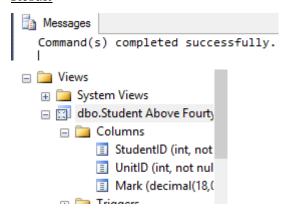
CREATE VIEW [Student Above Forty Marks] AS

SELECT StudentID,UnitID,Mark

FROM tblGrade

WHERE Mark>40

Result

```
Messages
  Command(s) completed successfully.

Views
  System Views
  dbo.Student Above Forty
    Columns
      StudentID (int, not
      UnitID (int, not nul
      Mark (decimal(18,0
    Triggers
```

Calculate minimum mark achieved any student
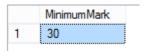
Self-Evalution

To calculate minimum mark by any student, min() aggregate function is used. Here, where clause is not required.

Solution

select MIN(Mark) as MinimumMark from tblGrade

Result

```
(1 row(s) affected)
```

	MinimumMark
1	30

List all the teachers who teaches units of BTTM
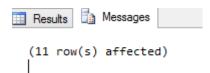
Self-Evalution

Data from three different table is required to be retrieved to solve the problem above. First, Program BTTM can only be filtered with help of Program table. Then, teachers can be filtered in teacher table. Now to make proper relation between these to table, unit table is required since both program and teacher is linked with unit table.

Solution

select t.Name,t.Designation,u.UnitID,t.Email,t.Address,t.Contact From tblTeacher T, tblUnit U, tblProgram P where U.TeacherID=T.TeacherID and P.ProgramID=U.ProgramID and P.Name='BTTM'

Result

Results Messages

```
(11 row(s) affected)
```

	Name	Designation	UnitID	Email	Address	Contact
1	Phoebe	Permanent	113	eu@ln.edu	Wanaka	9897633144
2	Noelani	Permanent	115	commodo@ametomarelectus.ca	Opglabbeek	9817333555
3	Fitzgerald	Permanent	119	pede.et.risus@nibhQuisquenonummy.co.uk	Tsiigehtchic	9829560874
4	Idola	Permanent	122	ipsum.leo@etmalesuadafames.edu	Alandur	9875504054
5	Melodie	Permanent	124	rutrum@Phasellus.org	Attigliano	9827485911
6	Xantha	Contract	125	at.risus.Nunc@vulputate.ca	Castello Tesino	9822421182

List the guardians of student from BBA

Self-Evaluation

To list out guardians of student from BBA, first student from BBA need to be filtered. To do so, select command is required to retrieve data from student, st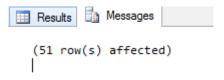udent status, guardian and program table. Program name BBA is filtered by Program.Name= 'BBA' then information about guardian is

displayed. To maintain the relationship, Student.StudentID= Studying.StudentID and Student.GuardianID=Guardian.GuardianID and Program.ProgramID=Studying.ProgramID is done.

Solution

Select G.GuardianID,G.Name, S.StudentID,S.Name,ST.ProgramID,ST.Semester from tblGuardian G, tblStudent S,tblStudying ST, tblProgram P where S.GuardianID=G.GuardianID and S.StudentID=ST.StudentID and P.ProgramID=ST.ProgramID and P.Name='BBA'

Result

	GuardianID	Name	StudentID	Name	ProgramID	Semester
1	1195	Orla	1007	Caesar	1003	First
2	1161	Clementine	1011	Lamar	1003	Second
3	1108	Karleigh	1016	Cameron	1003	Third
4	1061	Cassady	1017	Carol	1003	Fourth
5	1065	Fiona	1021	Maite	1003	Fifth
6	1052	Lucian	1022	Ira	1003	Third

Update Name of Department head of IT
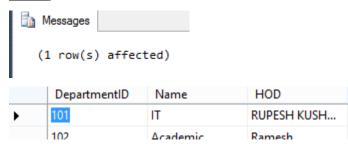
Self-Evalution

To Update row form department table, update statement is required to be used. To filter out department of IT where clause is used.

Solution

Update tblDepartment set HOD='RUPESH KUSHWAHA' where name='IT'

Result

	DepartmentID	Name	HOD
▶	101	IT	RUPESH KUSH...
	102	Academic	Ramesh
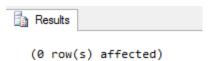
Update the mark gained by student ID 1003 in Unit Procedural Programming to 68

Update tblGrade set Mark=68 where StudentID=1003 and UnitID=(select UnitID from tblUnit where Name='Procedural Programming')

Result

Results

(0 row(s) affected)

Update Credit hour of all Units to 20
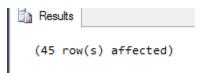
Self-Evalution

Again to update credit hour from Unit table, update statement is used. No where clause is required as there is no condition to filter.

Solution

update tblUnit set CreditHour=20

Result

Results

(45 row(s) affected)

UnitID	Name	CreditHour	TeacherID	ProgramID
101	Procedural Pro...	20	101	1001
102	Computing	20	102	1001
103	Web Desing	20	103	1001
104	Math	20	104	1003
105	Event Driven	20	105	1001

List all students whose name starts with "A"

Self-Evaluation
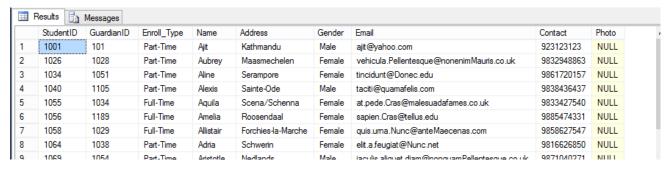
To solve the above problem select statement is required. Since, condition is to filter student whose name starts with 'A', 'Where' clause with like command is required to be used.

Solution

select * from tblStudent where Name like 'A%'

Results Messages

(21 row(s) affected)
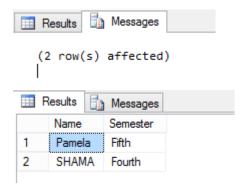
List all students from BTTM whose names ends with "A" and from Biratnagar

Self-Evolution

Here, select statement is used on join of tables student, student status and program is done. Data is filtered with Program name BTTM and like '%A'. relationship need to be maintained in query such as tblstudent.StudentID= tblstudying.StudentID.

Solution

select S.Name,ST.Semester from tblStudent S, tblStudying ST, tblProgram P where S.StudentID=ST.StudentID and ST.ProgramID=P.ProgramID and P.Name='BTTM' and S.Name like'%A' and S.Address='Biratnagar'
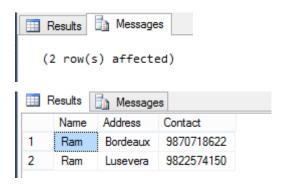

Find all students whose name is "Ram" and is in second semester

Self-Evaluation

Data is required to be retrieved from tables student and student status (studying) since, Name of student is stored in Student table and status of semester is stored in studying table. To maintain dependencies, studentID from both tables are compared. 'Where' clause on Name 'Ram' and semester 'Second' is used for filtering required information.

Solution

Select S.Name,S.Address,S.Contact from tblStudent S, tblStudying ST where S.StudentID=ST.StudentID and S.Name='Ram' and ST.Semester='Second'
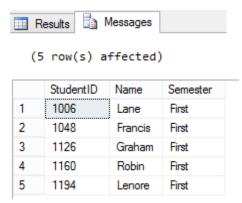
List Student who got less than 40 in first semester

Self-Evaluation

Student data is retrieved from student table where as mark and semester is filterer in grade and student status table respectively. Relationship should be maintained by comparing similar column in different tables. Then Where clause is used to filter semester= First and Mark<40.

select S.StudentID,S.Name,ST.Semester from tblStudying ST,tblStudent S, tblGrade G where St.StudentID=S.StudentID and ST.Semester='First' and S.StudentID=G.StudentID and G.Mark<40

Update all student of First-Semester to Second-Semester

Self-evaluation

To update all student from first semester to second semester, student status (studying) tales should be updated using while clause to filter student from first semester only.

Solution

Update tblStudying set Semester='Second' where Semester='First'

List all Teacher whose salary is above 25000

Self-Evaluation

To list all teachers with salary above 25000, select statement is required. Where-clause is used with (greater than) > operator for filtering rows in teachers table above 25000.

Solution

select * from tblTeacher where salary>2500
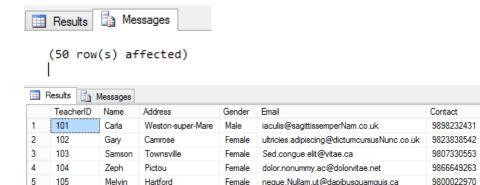
Update the salary of all teacher who is Permanent by 5%

Self-Evaluation

Update statement is required to update teachers table. With Where clause, rows are filtered with Designation= 'Permanent' and salary is set to 1.05%.

Solution

Update tblTeacher set salary=Salary+(salary*0.05) where Designation='Permanent'

Count female teacher in the college
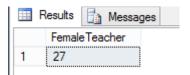
Self-Evaluation

Count() aggregate function is required to calculate number of rows after filtering teachers table with female gender. Count() function is used with TeachersID.

Solution

Select COUNT(TeacherID) as FemaleTeacher from tblTeacher where Gender='Female'

Calculate the minimum salary of a teacher(s) and increase it by 10%

Self-Evaluation

First row in teacher table with lowest salary is calculated. For this MIN() aggregate function is used. After that teachers table is need to be updated and set salary to 1.10% of the row which has minimal salary.

Solution

Update tblTeacher set Salary=Salary+(Salary*0.10) where Salary=(select MIN(salary) from tblTeacher)

Summary

In this report, twenty different problems are solved using different manipulation query tools. Different query were used for different purpose such as insert command for inserting data, delete command for deleting data and update for updating data in tables. This report also highlighted the advantage of different functions, clause and operators such as Where clause, >/< operator etc.

Task 3 [C]

Critically evaluate how meaningful data has been extracted through the use of query tools. [P3.3]
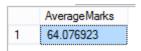
Introduction

This document evaluates how meaningful data has been extracted though the use of the query tools in previous problem solving task. For this result from different queries such as select, update, aggregate function join is compared and evaluated with the result we get from manual calculation. This will evaluate whether if data extracted from query is correct or not.

Result From Aggregate Function

Calculate the average mark of the BHM students
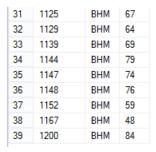
Result From Query

Evaluation

Average mark for BHM student in above problem is calculated using AVG () aggregate function. To evaluate the correctness of the result first the entire student From BHM and their Mark is retrieved using Select statement on join of tblStudying, tblPrograms and tblGrade.
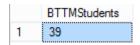
```
Select ST. StudentID, P.Name, G.Mark from tblStudying ST, tblGrade G, tblProgram P where
G.StudentID=ST. StudentID and P.ProgramID=ST. ProgramID and P.Name='BHM';
```

31	1125	BHM	67
32	1129	BHM	64
33	1139	BHM	69
34	1144	BHM	79
35	1147	BHM	74
36	1148	BHM	76
37	1152	BHM	59
38	1167	BHM	48
39	1200	BHM	84

Sum of the Marks of all 39 students after manual calculation was 2499. Hence average marks of BHM student is 2499/39=64.076923 which is exactly same as the result found after using Avg (Mark). This shows data has been extracted correctly.

Calculate the number of student who joined BTTM
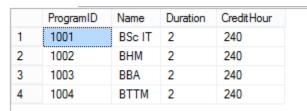
Result from Query

Evaluation

Firstly, ProgramID of BTTM is found using select statement on Program table then the entire student studying on that program from studying table is counted manually to compare the result.

Select * from tblPrograms;

	ProgramID	Name	Duration	CreditHour
1	1001	BSc IT	2	240
2	1002	BHM	2	240
3	1003	BBA	2	240
4	1004	BTTM	2	240

Above table shows ProgramID of BTTM is 1004. Now another query is executed on studying table and filtering ProgramID 1004.
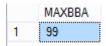
Select * from tblStudying where ProgramID=1004

34	1169	1004	Sixth
35	1187	1004	Fifth
36	1194	1004	Second
37	1195	1004	Second
38	1198	1004	Fifth
39	1201	1004	Sixth

Now manually counting student from the table we get 39, also shown in figure above. This result is exactly as same as result we got from count () aggregate function which shows data has been extracted correctly.

Calculate the maximum mark gained by full-time student of BBA

Result From Query

	MAXBBA
1	99
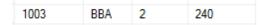
Evaluation

To calculate the result manually first marks of student from full time student of BBA is required. For this first ProgramID of BBA is found using select query on program table.

Select*from tblProgram where Name='BBA'

This showed ProgramID for BBA is 1003.

1003	BBA	2	240

Now using another query is required to find marks of student form ProgramID 1003 using a join. And by using ORDER BY DESC key word, maximum mark can be found.

Select G.Mark from tblGrade, tblStudying ST where G.StudentID=ST. StudentID and ST. ProgramID=1003 order by G.Mark DESC

	Mark
1	99
2	97
3	95
4	93

Now as the marks are in descending order, topmost mark is maximum mark which is 99. This result is exactly as same as result we got from using MAX () aggregate function. This shows data has been extracted correctly.

Result from joining table

List all the teachers who teaches units of BTTM

Result from Query

	Name	Designation	UnitID	Email	Address	Contact
1	Phoebe	Permanent	113	eu@ln.edu	Wanaka	9897633144
2	Noelani	Permanent	115	commodo@ametornarelectus.ca	Opglabbeek	9817333555
3	Fitzgerald	Permanent	119	pede.et.risus@nibhQuisquenonummy.co.uk	Tsiigehtchic	9829560874
4	Idola	Permanent	122	ipsum.leo@etmalesuadafames.edu	Alandur	9875504054
5	Melodie	Permanent	124	rutrum@Phasellus.org	Attigliano	9827485911
6	Xantha	Contract	125	at.risus.Nunc@vulputate.ca	Castello Tesino	9822421182

Evaluation

The ProgramID of BTTM is 1004 as found from using select * from tblProgram. Now units from 1004 program is filtered using where clause.
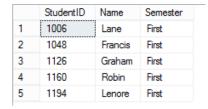
```
Select * from tblUnit Where ProgramID=1004
```

	UnitID	Name	CreditHour	TeacherID	ProgramID
1	113	Nunc Sed Pede Corporation	20	113	1004
2	115	Id Sapien LLC	20	115	1004
3	119	Magnis LLC	20	119	1004
4	122	Sociis Natoque LLP	20	122	1004
5	124	Adipiscing Corp.	20	124	1004
6	125	Placerat Cras Inc.	20	125	1004
7	130	pede.	20	130	1004
8	133	cursus	20	133	1004
9	137	sit	20	137	1004
10	140	interdum.	20	140	1004
11	142	augue	20	142	1004

Now manually checking the entire teacher in teacher table by the TeacherID we found in above table we get same Phoebe, Noelani etc. This is same as result we got from using join query to list teachers of BTTM units. This evaluation shows data has been extracted correctly using join query.

List Student who got less than 40 in first semester

Result From Query

	StudentID	Name	Semester
1	1006	Lane	First
2	1048	Francis	First
3	1126	Graham	First
4	1160	Robin	First
5	1194	Lenore	First

Evaluation

For this firstly, all information from grade table is retrieved using ORDER by key word on MARK to list out students with less marks than 40 manually.

	StudentID	UnitID	Mark
5	1006	127	32
6	1153	114	32
7	1138	114	32
8	1143	120	34
9	1160	110	34
10	1090	136	36
11	1059	125	36
12	1197	143	36
13	1196	134	37
14	1111	126	37
15	1126	130	38
16	1128	124	38
17	1033	109	38
18	1194	131	38
19	1168	138	39
20	1032	115	39

Now from manually comparing marks student ID form table above and studying table we get student ID 1006, 1048, 1126 etc. Again manually checking these student ID in Student table we get lane, Francis, graham etc. which is same as we got from join query using < operator. This evaluation shows data has been extracted correctly.

List all Teacher whose salary is above 25000

Result from Query

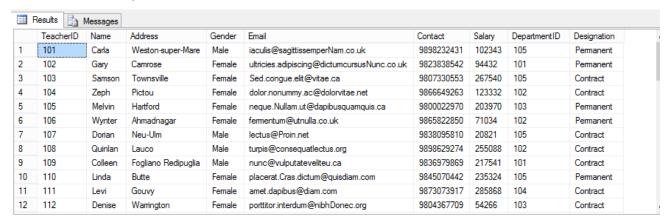|	TeacherID	Name	Address	Gender	Email	Contact	Salary	DepartmentID	Designation
1	101	Carla	Weston-super-Mare	Male	iaculis@sagittissemperNam.co.uk	9898232431	102343	105	Permanent
2	102	Gary	Camrose	Female	ultricies.adipiscing@dictumcursusNunc.co.uk	9823838542	94432	101	Permanent
3	103	Samson	Townsville	Female	Sed.congue.elit@vitae.ca	9807330553	267540	105	Contract
4	104	Zeph	Pictou	Female	dolor.nonummy.ac@dolorvitae.net	9866649263	123332	102	Contract
5	105	Melvin	Hartford	Female	neque.Nullam.ut@dapibusquamquis.ca	9800022970	203970	103	Permanent
6	106	Wynter	Ahmadnagar	Female	fermentum@utnulla.co.uk	9865822850	71034	102	Permanent
7	107	Dorian	Neu-Ulm	Male	lectus@Proin.net	9838095810	20821	105	Contract
8	108	Quinlan	Lauco	Male	turpis@consequatlectus.org	9898629274	255088	102	Contract
9	109	Colleen	Fogliano Redipuglia	Male	nunc@vulputateveliteu.ca	9836979869	217541	101	Contract
10	110	Linda	Butte	Female	placerat.Cras.dictum@quisdiam.com	9845070442	235324	105	Permanent
11	111	Levi	Gouvy	Female	amet.dapibus@diam.com	9873073917	285868	104	Contract
12	112	Denise	Warrington	Female	porttitor.interdum@nibhDonec.org	9804367709	54266	103	Contract

Evaluation

Salary of the each teacher can be listed in order just by using ORDER BY key word which can ease the manual count of teacher who got above 25000 Salary.

Select * from tblTeacher ORDER BY Salary

	TeacherID	Name	Address	Gender	Email	Contact	Salary	DepartmentID	Designa
1	133	Venus	Ururi	Male	lorem.ac@orci.co.uk	9873875783	21577	103	Contrac
2	107	Dorian	Neu-Ulm	Male	lectus@Proin.net	9838095810	22903	105	Contrac
3	112	Denise	Warrington	Female	porttitor.interdum@nibhDonec.org	9804367709	54266	103	Contrac
4	135	Emmanuel	Villata	Male	et@fames.co.uk	9836064711	56256	104	Permar
5	130	Chastity	Pangnirtung	Male	mi.felis@sitametrisus.co.uk	9838004050	59938	101	Contrac
6	143	Rana	Kansas City	Female	ipsum@eleifendnec.net	9899816654	62665	104	Contrac
7	140	Micah	Caprino Bergamasco	Male	in@nonvestibulumnec.org	9807427572	63818	101	Permar
8	122	Idola	Alandur	Female	ipsum.leo@etmalesuadafames.edu	9875504054	67132	105	Permar
9	106	Wynter	Ahmadnagar	Female	fermentum@utnulla.co.uk	9865822850	74586	102	Permar
10	123	Aidan	Ch?telet	Female	fames.ac@duiFuscealiquam.net	9892933969	77026	103	Contrac
11	120	Charlotte	Buckie	Female	egestas.Aliquam.fringilla@adipiscingligula.edu	9875544701	93862	105	Contrac
12	142	Veronica	Belgrave	Female	ac.orci.Ut@Aliquam.com	9887858508	95140	103	Contrac

Now manually listing teacher with above 25000 salaries we get teachers Denise, Dorian, Venus, Micah etc. which is similar to the list we got from use for join query during problem solving. Evolution shows data extracted are correct.

Critical Evaluation

This document evaluated result from several queries executed to solve problems in previous tasks. Evaluation was done on whether use of different key words, clauses, aggregate functions and join extracted correct data or not. After comparing result from queries in previous problem solving and manual calculation it is found that result are exact in both cases. It is found that use of aggregate function such as MAX (), COUNT (), AGE () etc. and <,> etc. operator helps to find correct data and makes process simple and quicker. Manual data finding was more time consuming and were stressful. This critical evaluation shows data has been extracted correctly via use of SQL DML queries.

Task 4 [A]

Critically review and test a relational database system. [P4.1, D1]

Introduction: Database Testing

Purpose of database Testing

Machanic (2008) writes testing is software is very important as it makes sure users do not find errors while using the system. The main purpose of the testing is to find errors in the system and repair them. Hence it makes sure system has no error/bugs that are to be debugged/ repaired. Here are some of the key benefits of database testing.

1. To understand the consistency of the developed database system.
2. To check test bugs in database and objects in database.
3. To check if database matches system specification i.e. schema diagrams etc.
4. To check usability database system
5. To check database system supports queries relationship, dependencies etc.

Categories of testing

Black box testing

Black box testing is type of testing that does not requires knowledge of internal structure of the system. Tester does not have access to the system code and only performs testing on UI. Testers provide input and analyze the output with expected output without any proper knowledge of how system is working internally. Some of the key benefits of this type of testing involve:

1. Access to code is not required
2. Test in done from USER prospective
3. Tester can perform this test with even with no proper knowledge or idea about how system is build.

Black box testing has its limitations as tester has only little knowledge about system's architecture test is not efficient itself. In addition to that test cases are difficult to design.

White box testing

White box testing on the other hand requires proper knowledge about internal structure and requires access to the codes. It is detailed testing of internal logic and structure. This type of testing is also described as glass testing or open testing. Some of the benefits of white box testing are:

1. Since has access and knowledge of codes, tester can design test cases effectively
2. Test result helps to optimize the code. Extra and unrequired codes are removed making system better.

Database testing is essential phase in software development. This test document tests the developed database system to analyze the efficiency of the system. Different elements of database is tested and logged in report during this test. Some of elements that are tested are listed below:

- Tables
- Queries
- Relationships
- Usability's of database
- Constraints etc.

Test log for the performed test is attached in separate page below.

References

Machanic, A. (2008) *Expert SQL Server 2008 Development* USA: N.A.

Task 4 [B]

Create documentation to support the implementation and testing of relation database system. [P4.2]

Introduction

The purpose of this document is to provide guidance and support for the implementation and support for the developed database system. To support and maintain the database system technical user can follow this document as technical guidance. It also covers support for maintaining and testing the system.

Support on Implementation

This section of the document provides support on implementation of database. Implementation support section covers creating a database, using it, creating a table, dropping a table, alternating a table and maintaining relationship (referential integrity) etc.

Creating database

1. Start SQL SERVER management system
2. Click on New Query
3. Now type query statement and click on execute to build database.

   Create Database <database_name>

Use Database

To create database objects in the created database first database in need to be make active. For this USE statement is used.

1. Type query statement and execute.

   Use <database_name>

   (Alternatively, Database can be manually selected from drop down box name database)

Drop Database

Sometimes database is needed to be deleted. For this Drop statement is used.

1. Type query statement and execute.

   Drop <database_name>

Creating Table

Now database is created and is used, relational table is created using Create statement. For this following syntax is used.

CREAT <table_name> (attribute 1, attribute 2…);

Example:

Create table tblStudent(StudentId int, Name nvarchar(40));

Above query statement creates table named tblStudent with StudentId and Name columns.

Defining Data Types

Data type in query statement is followed by attribute name while creating a table. It defines what type of data that attribute supports. For example:

StudentId INT,

Name NVARCHAR (50),

Adding Keys

Keys in query statement are followed by data type while creating a table. It can also written followed by constrain name identity. For example:

Constraint PK_SID Primary key (StudentId);

Adding Domain Constraint

Data type itself is a domain constraint which forbids data values other than that specific data type. Additional domain constrain can be used such as not null. Not null prevents null value and is followed by data type in create statement. For example:

Name NVARCHAR (50) not null;

Managing Referential constraints

To maintain relation between two or more tables referential constraint is used. Primary key and foreign key is used to establish and maintain relation. For example:

Create table tblGrade (

StudentId int not null,

Marks decimal not null,

Constraint FK_SID Foreign key (StudentId) references tblStudent (StudentID));

Here in above create statement relation between tblGrade and tblStudent is established by referencing studentId from tblStudent table.

Altering a Table

For some reason if a table is needed to be modified to insert or delete a column, alter statement is used. Syntax for the alter statement is as follows:

Adding a column:

ALTER TABLE <table_name> ADD <column_name> data_type

Deleting a column:

ALTER TABLE <table_name> DROP COLUMN <column_name>

Changing a column's data type:

ALTER TABLE <table_name>ALTER COLUMN <column_name> data_type

Dropping a table

To delete a table from database Drop statement is used.

DROP TABLE <table_name>

Support on Testing

This section of document covers how testing of different elements of database is done. Different elements of database are tested during testing phase such as tables, relations, queries etc. This also provides supports on carrying out more depth tests on database.

Testing Database

Usability, read/write capability and max size of database is tested in this phase. To find the test result, different queries are used such as sp_helpdb; this query statement gives information about database. Similar approach can be used to test owner of database.

Testing Table

For testing tables, first number of tables in database is counted to ensure database has exact number of entities according to ER-diagram. Then attributes of tables are tested using sp_help <table_name> statement. Testing of table also include inserting more values than the number of column.

Testing Relation

To test relation foreign key and primary key relation (parent-child) relation is tested. For example what happens when data from parent is deleted when child still has that value? Or what happens when a new data is inserted in child when that value is not available in parent column. This type of test checks relation health in database system.

Testing Queries

Testing of queries checks weather database supports SQL queries. Different queries are executed to find the result. Queries can be performed on joins, tables, functions, view of database.

Creating Test logs

To analyze the test result it is recorded in a test log. This test log can be in table format given below.

S.N.	Date	What was tested	Test Data	Expected output	Actual output	Remarks

Task 4 [C]

Create user documentation for a developed relation database system. [P4.3, M3]

Introduction

This user documentation covers user guidance to use different feature of database. This document supports user to store, update, delete, and finding data from database. Fundamental usage of database is covered in this document.
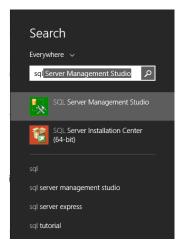
User Documentation

Start SQL server management Studio

1. Go to C:→Program Files (x86)→Microsoft SQL Server→110→Tools→Binn

2. Double click on ManangementStudio folder

3. Open Ssms.ext (SQL SERVER MANAGEMENT SYSTEM)

Alternative:

1. Press Windows + S in Windows 8.1

2. Type SQL server management studio→ Click on SQL server Management

Logging into SQL server

1. After starting up SQL server management studio, Login windows should open.

2. Now either choose Windows Authentication (Figure 1) or SQL server Authentication (Figure 2)

3. Provide Password and Username. → Connect

Setting up Query Environment

1.   Right click on database and select   NEW query

Finding all information from a table

To find all information from a table Select Statement is used with no 'Where' clause.

Example:

Select * from tblStudent;

Above query lists all information from tblStudent table.

Filtering unwanted rows

To filter unwanted information from the search Where-clause is used.

Example:

Select * from tblStudent where StudentId=1009;

	StudentID	GuardianID	Enroll_Type	Name	Address	Gender	Email	Contact	Photo
1	1009	1045	Part-Time	SHAMA	Biratnagar	Male	faucibus@PhasellusnullaInteger.edu	9864976344	NULL

Used statement above filters Student(s) whose id is 1009. Other filter action can be done similarly in different tables.
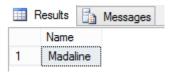
Retrieving information from specified columns only

To retrieve information from specific columns only, select column's name is followed with select statement. Multiple columns can be selected with comma (,).
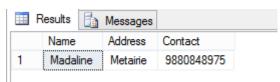
Example:

Single Column:

Select Name from tblStudent where StudentId=1008;

	Name
1	Madaline

Multiple Columns:

Select Name, Address, Contact from tblStudent where StudentId=1008;

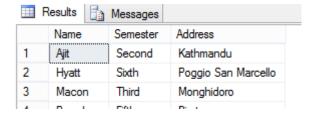|	Name	Address	Contact
1	Madaline	Metairie	9880848975

Retrieving data from more than 1 table

To retrieve data from multiple tables, data is retrieved by forming join and maintaining child parent relationship.

Example:

Select tblStudent.Name, tblStudying.Semester, tblStudent.Address from tblStudent, tblStudying where tblStudent.StudentId=tblStudying.StudentId

	Name	Semester	Address
1	Ajit	Second	Kathmandu
2	Hyatt	Sixth	Poggio San Marcello
3	Macon	Third	Monghidoro

Updating a Row in a table

To update value(s) in row update statement is used. If value of a specific row is needed to be modified, where-clause is used.

Example:

Update tblStudent set Name='Atut' where StudentId=1001;

Above statement updates row in student table and set name to Atut in which student id is 1001.

Deleting a row

To remove a row from a table Delete statement is used with Where clause to filter wanted row.
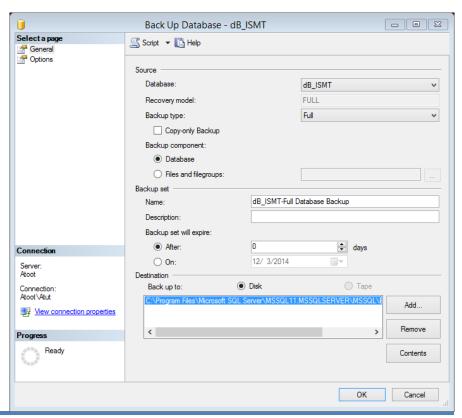
Example:

Delete from tblStudent where StudentId=110;

This query deletes students whose id is 110.

Backing up Database

Database back up is an important step to ensure data is safe even in case of system failure. Data is backed up in different location or in a file which can be transferred to different location. This back up can be restored when required. Most recent back up file is restored in case of system failure or data loss. To perform database back up follow these steps:
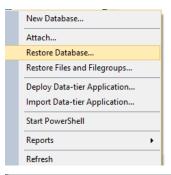
1. Right click on database
2. Select Tasks→ Back up…
3. Select Back up type, Location
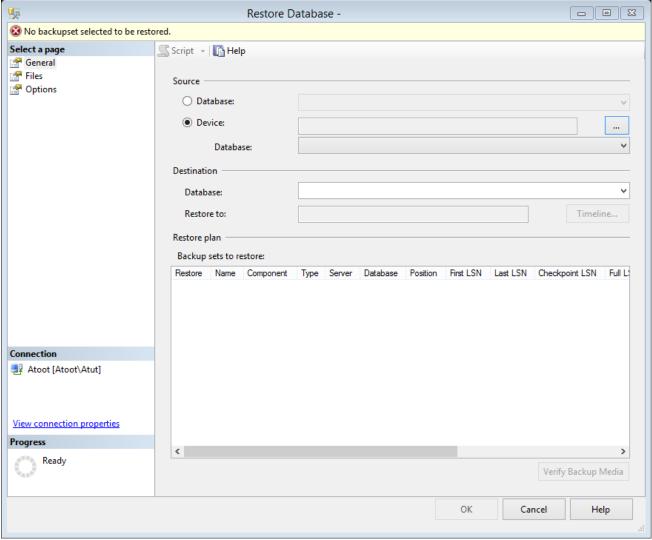4. Click on ok
5. Once process in finished back up is done.

Restoring Database

To restore from back up file, follow these step:

1. Right click on databases

2. Select database restore

3. Now in new window select device and locate database back up.

4. Click on ok. Done

Task 4 [D]

Explain how verification and validation has been addressed. [P4.4]

Introduction: Data verification and validation

In software development, data validation and verification is process for making sure that it processes only valid and useful data. It checks the weather data generated is meaningful, consistence and satisfies the system specification. There are set of rules that are implemented to ensure correct data are passed to database. Some of the examples of validation testing are data type validation, range-constraint validation etc.

- Data type validation

This type of validation ensures database follows data type's specification.

- Range constraint

This validation checks weather user can input data above or below the data range.

For example, data validation marks column in grade table only allows numeric data.

Data validation can be implemented via business logics in business layer of software or through integrity rules.

Validation and Verification in developed database system

S.N	Validation/verification Type	Testing method	Expected result	Actual Result
1.	INT data type test	Inserting larger data than INT data type capacity i.e. larger than 2,147,483,647 i.e. insert into tblGrade (StudentID,UnitID,Mark) values (1001,101,99999999999999999999999999)	Should throw error message disallowing to insert new row.	Threw error message while executing query saying arithmetic error.

```
insert into tblGrade (StudentID,UnitID,Mark) values (1001,101,999999999999999999999999999
```

100 %   ▾   ‹

📄 Messages

```
Msg 8115, Level 16, State 8, Line 1
Arithmetic overflow error converting numeric to data type numeric.
The statement has been terminated.
```

Exmplanation

Here int data type validation test is done. For this much larger data than int capacity is tried to insert into int data type column in grade table. It is expected to throw some error. Indeed, while executing the query, statement was terminated showing error saying arithmetic overflow.

2	nVarchar(n) data type test	Inserting larger data than n specified in nvarchar(n) i.e. 20 character is inserted in Enroll_Type column of tblEroll_Type table Insert into tblEnroll_Type values ('ABCDEFGHIJKLMNOPQRSTUVWXYZ','test')	Should throw error message disallowing to insert new row.	Error messages appeared saying string would be truncated.

📄 Messages

```
Msg 8152, Level 16, State 4, Line 1
String or binary data would be truncated.
The statement has been terminated.
```

Explanation

In this test, test is done on how system reacts when input data exceed the specified data type of column. Test is done by inserting 26 characters in 15 character nvarchar which is capable of storing only 15 characters. This results error message saying string would be truncated as it cannot hold input data. This show data type and their size has some meaning and its working.

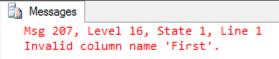| 3	10/30/2014	Not NULL KEY:	Insert statement with no value for not null column. Insert into tblGrade (Unit id, mark) values (1001, 99)	Should throw error message. Does not allow inserting new row.	Threw error message null value not allowed.

```
 Messages
   Msg 515, Level 16, State 2, Line 1
   Cannot insert the value NULL into column 'StudentID', table 'dB_ISMT.dbo.tblGrade'; colum
   The statement has been terminated.
```

Explanation:

Not null key is domain constrains that makes sure data in that column while inserting or updating a row cannot be null. To test the Not null domain constrain, null value is tried to be stored in column which is restricted with not null constraint. Using Insert into tblGrade (Unit id, mark) values (1001, 99) tries to leave null data in studentID column in tblGrade table. StudentID is not null hence it is expected to show error message and prevent to insert row. Test result after query execution shows inserting null value in not null column is restricted. Shows test is successful.

4	Data Type test: Insert Int value in nvarchar (n) field.	To test data type validation, different data type is inserted into a column that their definition. Query: tblSemeseter(First,'First semester')	Should throw error message.	Threw error saying invalid column name.

```
 Messages
   Msg 207, Level 16, State 1, Line 1
   Invalid column name 'First'.
```

Explanation:

While inserting new row in tblsemester, semester column has datatype nvarchar whichi accepts data with single quote like 'RAM'. But here query has no '' hence it is expected to throw error message. Upon execution, as expected error was thrown which shows database is well data type verified.

5	Data redundancy test: Insert duplicate row twice in table	To perform this validation test, same insert query is executed. Insert into tblSemeseter('First','First semester')	Should throw error message.	Threw error saying query violates primary key constraints.

```
 Messages
   Msg 2627, Level 14, State 1, Line 1
   Violation of PRIMARY KEY constraint 'PK_Semester'. Cannot insert duplicate key in object 'dbo.tblSemester'. The d
   The statement has been terminated.
```

6	Referential Integrity Check: Delete from Parent while child still has the data	Student is tried to be deleted from student table while it still exists in other child table such as grades, studying tables. Query: Delete from tblStudent where StudentID=1001;	Should terminate the process and throw error meessge.	Error message displayed saying violation of foreign key.

```
Messages
  Msg 547, Level 16, State 0, Line 1
  The DELETE statement conflicted with the REFERENCE constraint "FK_StudentID". The conflict occurred in database "
  The statement has been terminated.
'
```

7	Referential Integrity Check: Insert into child while parent do not have the data	Query: Insert into tblGrade values (1000001,1004,41)	Should throw error message.	Error message was displayed.

```
Messages
  Msg 547, Level 16, State 0, Line 1
  The INSERT statement conflicted with the FOREIGN KEY constraint "FK_StudentID@2". The conflict occurred in databas
  The statement has been terminated.
```

Explanation:

To check referential integrity, data from parent column is tried to delete and new data in child column in tried to insert. But due to dependencies and relationship in the table it was not allowed to violate foreign key rules. This shows database shows good referential integrity verification.

8	Range Verification Marks above 100 and below 0.	Higher mark is inserted into mark column of tblGrade table. Insert into tblGrade values (1004,108,109)	Should throw error message.	No error message was displayed. Instead, messaged said 1 row affected.

```
Messages
  (1 row(s) affected)
```

Explanation:

Above verification test shows database is failed to verify valid marks. Student marks above 100 and below 0 should not be allowed. This verification can be done in business layer later while developing UI.

Task 4 [E]

Explain how control mechanisms have been used. [P4.5]

Introduction

Control mechanism in database system ensures security, integrity and availability of a database system. Security mechanism detects and prevent unauthorized accesses to information while, integrity control prevents improper changes in information. Control mechanism controls allowing or disallowing of user actions on the databases and their objects. This document explains some of control mechanisms used in current system.

Control Mechanisms in developed database system

Access control

Access control is mechanism which allows or prevents user from accessing the database system. User, first needs to login to system using authentication process. Access control includes permission to creating, alternating or dropping a database or a database object. Access control also controls user permission to makes chances in database information such as inserting rows, updating, deleting, retrieving etc.

In current system after starting SQL server management system user must need to provide valid authentication credentials to gain access to database. Only user with permission to database is allowed to modify database. If user fails to provide valid login information, he/she will have no access to database. Create User Statement can be used to create a new user for a database.

Create View for security purpose

According to Microsoft (n.d.) creating view can work as security mechanism as it prevents direct access to table. It helps to hide confidential column. For example, in tblStudent if only name and Student ID is required to be shown to user, view can be created only including name and student id column.

Create view Student as select StudentID, Name from tblStudent;

This prevents user from accessing other confidential information of student.

Domain and Referential constrain

Other control mechanism used in current system is use domain and referential constraints. It controls the actions on database objects and data by implementing domain and referential rules in database.

This mechanism helps to prevent improper changes in database, for example:

- It ensures user cannot insert marks above 100 or less than 0.
- User cannot delete a unit if students are studying that unit.
- User can not add student to a new program is program does not exist in program table.

Future scope for control mechanism

Gran/Privilege

Gran statement is used for granting privilege on a resource to specific user or group (oracle, n.d.). Here resource can be a table, database, view or function etc. Additionally, User can be individual user of group of user. Privilege is rules to perform query statement such as delete, update, select, trigger etc. Some of the table privilege types are as follows:

1. Select-   Grant to perform Select statement
2. Update-   Grant to perform Update statement
3. Delete-   Grants to perform delete statement
4. Insert-   Grants to perform Insert Statement

Grant syntax is shown below:

 GRANT *Privilege-Type* **ON** *[Table or View]* **TO** *[User or Group]*

For example:

Grant SELECT ON tblStudent TO Counselor

In above grant statement, User counselor is given grant to perform select query of tblStudent table. Multiple privileges can be grant to a user using comma (,).

Trigger

Trigger is program (function or constraint) that automatically runs when certain event is occurred. Trigger is control mechanism that can perform different action based on event occurred. Trigger has three different components:

- Event: Trigger is activated when event is occurred. Event can be Insert, delete or update, Create, alter, delete
- Condition: Trigger is activated only when condition is true.
- Action: This section defines actions trigger perform upon activation

Trigger action can be of two types. In type on action takes place before event is executed. In other type, action takes place after event is executed.

Benefits of Using trigger:

- Assessing and Maintaining referential integrity
- Commanding security authorizations
- Checking invalid transactions

Syntax for trigger:

Create Trigger *<name>*

Before| After   Insert| Update| Delete On *<table_name>*

For Each Row | For Each Statement

When *<condition>*

Example

```
CREATE TRIGGER reminder2
ON Sales.Customer
AFTER INSERT, UPDATE, DELETE
AS
    EXEC msdb.dbo.sp_send_dbmail
        @profile_name = 'AdventureWorks2012 Administrator',
        @recipients = 'danw@Adventure-Works.com',
        @body = 'Don''t forget to print a report for the sales force.',
        @subject = 'Reminder';
GO
```

In above example, trigger is activated when Insert, update or delete is done on Sales.Customer. By this way trigger can control actions done in database management system.

Concurrency control and Recovery control

Modern database systems mostly used in multi-user environment. Users access database concurrently in these environments. Concurrency control is method to maintain conflict with simultaneous access of database by different users. Use of concurrency controls ensures, simultaneous transactions are co-ordinated and data integrity is maintained. For example, if a product in web store is brought by two users at same time, conflicts can generate their but concurrency controls handles the situation and sell the product to one user.

Recovery meanwhile ensures data in database does not get corrupted as result of system failure or misuse of software. The main function of the recovery control is to maintain fault tolerance of database.

Recovery Controls covers facing these issues:

1. Facing failures
   a. System failures
   b. Transaction failures
   c. Media failures

Summary

On summary, this document explained what and how control mechanisms are used in the current database system. SQL login feature is used for access control. Only valid users are allowed to access the database. For the data integrity, domain and referential constrains are used. Views are created to prevent users from direct access to tables. Document also explained other control mechanism that

can be implemented on database system such as grant and privilege, Triggers etc.

References

- Microsoft (n.d.) *Using Views as Security Mechanisms* Available: http://technet.microsoft.com/en-us/library/aa905180(v=sql.80).aspx Accessed [09/25/2014]

- Oracle (n.d.) *Grant Statement* Available: http://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqljgrant.html Accessed [09/25/2014]