PART 1A

A database management system (DBMS) will be created to record, access and store details of risks in the Information Services Section of Griffith University. The database is necessary for the following reasons below

1. To provide concurrent, distributed access to large volumes of data.

Currently data of risks are stored in a large spreadsheet containing nearly 30 columns with significant amounts of redundant information and the spreadsheet is potentially required to store hundreds of records. Every year the Risk and Compliance manager of Information Services receives approximately 50-100 risks from external audits as well as risks identified by other university stakeholders such as staff and students. The Risk and Compliance Manager then populates the spreadsheet with details of the risks. Periodic updates on the risks are supplied by various parties including the system owner(s) and the contact person(s). Internal Audit as well as the CTO and PVC (INS) also need to have regular access to the updated spreadsheet.

From the description above it appears that a spreadsheet is problematic as there are different versions of the spreadsheet floating around and it requires the Risk and Compliance Manager to personally update the spreadsheet all the time. By creating a database staff members can have the permission to update risks which they are responsible for.

2 + 3. To provide a relational data model and a powerful, uniform language for querying and updating data (SQL).

A relational data model represents data in tuples grouped into relations. It allows users to query and update the database with a powerful uniform language (SQL) commonly used in many DBMS including Oracle and Derby. SQL is also a declarative language that allows users to directly state what the database contains and what data needs to be extracted without having to specify how to do it. The DBMS is responsible for describing data structures for storing the data and the retrieval procedures for answering queries.

4. To allow powerful optimisations for efficient query evaluation (indexing, query transformation).

A DBMS can perform query transformations, a set of techniques used by the optimizer to rewrite a query and optimise or improve it. For example, determining the join order and join strategy. A DBMS can also place queries on tables to speed up queries. By default an index is normally placed on the primary keys of the table but in certain situations hash indexes can also be placed on the joins of tables to speed up the query.

5. To ensure data integrity within single applications.

Data integrity falls into these categories entity, domain and referential integrity. Each of these will be discussed in turn.

Entity integrity ensures that each table within the database has a primary or composite key that serves as a unique identifier for rows in the table. The primary or composite key cannot occur in any other row in the table. This prevents duplication of records.

Domain integrity is a constraint checking mechanism for columns to ensure that the columns only accept valid data entries on insertion or update. Examples of data Domain Integrity constraints for this project are:

- Primary key Risk_id cannot be null
- Date_raised- to accept dates only
- Risk_id in specific format only that is INS followed by up to 6 numbers eg “INS73”
- Certain columns for example-
    - Risk_category can only accept specific values eg “Operational”, “ICT”, “Project”)
    - Risk_rating can only be “high”, “medium” or “low”

Referential integrity preserves the defined relationships between tables when records are entered or deleted and it is based on relationships between foreign keys and primary keys. It ensures that key values are consistent across tables. An example of how referential integrity can assist this project is illustrated below.

Currently data is stored in spreadsheet which creates problems if the risk framework is updated. For example risk likelihood of low and a risk impact of high gives rise to a medium instead of a high risk. The spreadsheet will have to be checked manually update risk rating and process may potentially give rise to data inconsistency and errors. By using a relational data model redundancy is also eliminated as a risk rating does not have to be stored for each risk item but can be obtained by performing a lookup from the rating table using the risk likelihood and risk impact.

Routine data backup and restore procedures are necessary for preserving the database integrity and restoring the database after power failures or bad sectors in disk.

6. To ensure data integrity across multiple concurrent applications

Multiple users may seek to access or alter data simultaneously. An optimistic approach is taken to handle multiple accesses to the database as it is unlikely that multiple users will be updating same field in a record. In this approach a database transactions use data resources without applying locks to those resources first. Before committing the transaction, the database verifies that no other transaction has modified the data read. If conflicting modifications are revealed then the committing transaction is cancelled and rolled back.

Potential for the operational data to be the source of analytics.

University and INS collects statistics on the following

- The number of finalised/unfinalised risks each year
- number of risks in each category
- how efficiently are risks addressed
- details for forecasting and making predictions of risk profile

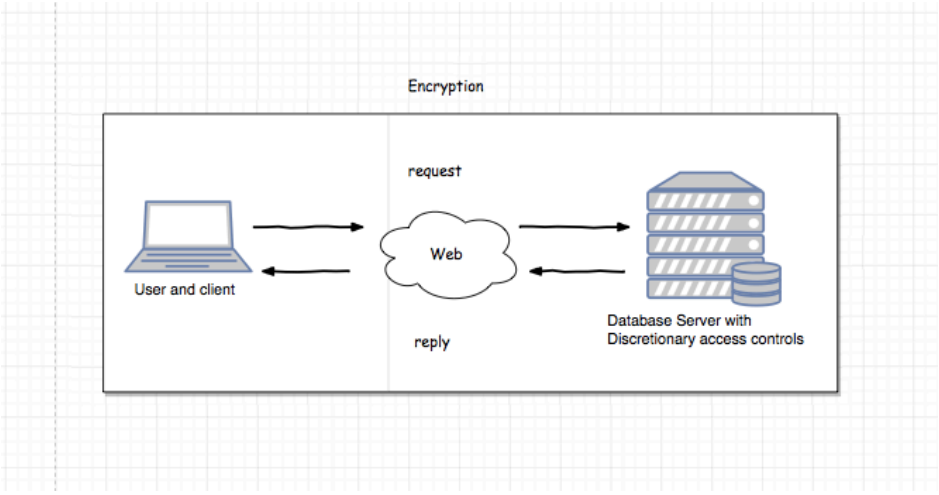
Statistics can also be collected on how efficient are data accessed/updated to monitored the database and identify anomalies.

PART 1B

Derby is sufficient for this project as the expected number of users accessing the data at any given moment is not significant and the size of the data stored is relatively small and can fit into one server. This database is supported by two major platforms used by staff, MAC and Windows.

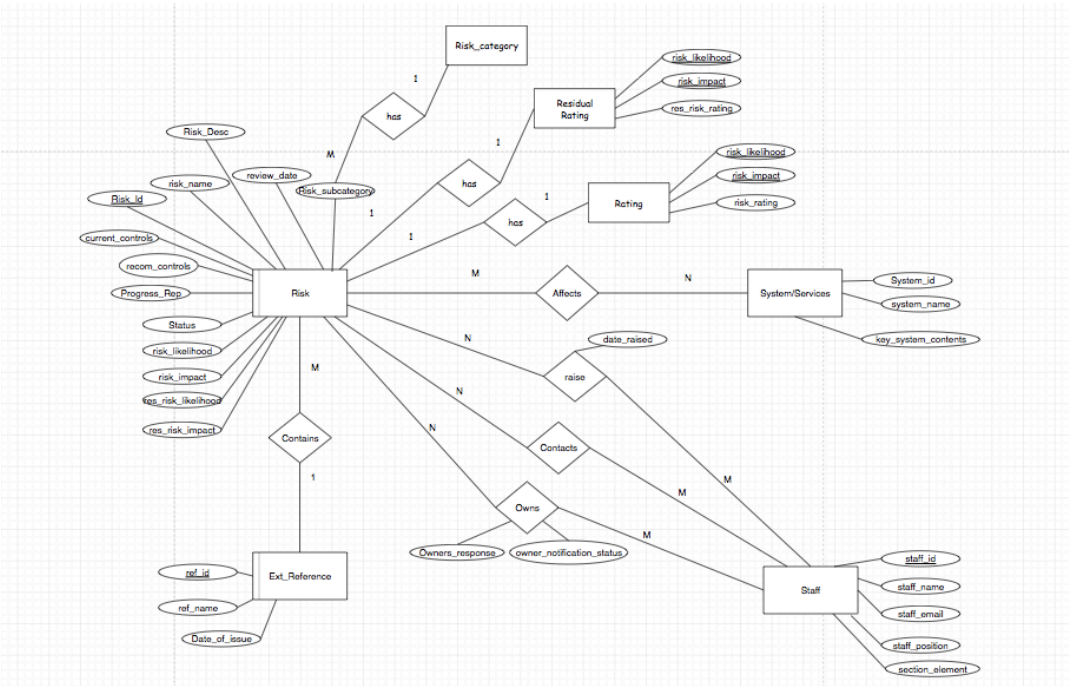
Although three tier architecture offers greater performance and security over a two tier architecture model, a two tier architecture model is requires less resources to implement and sufficient for this small scale project. In a two tier architecture the client requests resources from the database server and the database server responds directly to the request using its own resources. The security of the two tier architecture can be enhanced by encrypting data transferred in the web between the client server and the database server. Discretionary access controls are also employed to grant users with specific privileges or access rights to tables and views.

Figure 1: Two tier architecture



PART 2 – Conceptual Design

ER diagram



Typical Queries

Query to select risks with risk rating of high

```
SELECT R.Risk_Name
FROM Risk R, Rating S
WHERE R.risk_likelihood = S.risk_likelihood
AND R.risk_impact = S.risk_impact
AND R.risk_rating = “High”
```

Query to return the names and owners of open risks with reviews due within the next month

```
SELECT R.Risk_name, O.Staff_name
FROM Risk R, Owns O
WHERE R.Risk_id = O.Risk_id
AND R.review_date BETWEEN 3/4/2016 AND 30/4/2016
AND R.Status = “open”
```

Examples of Updates and Insertions

Insertion to create a new risk

```
INSERT INTO risk VALUES (‘INS103’, ‘Student Portal’, ‘possible attack’, …….);
```

Updates to the details of risks are performed regularly. For example:

```
UPDATE Risks
SET Risk_Desc = “place description here”
WHERE Risk_id = “123”
```

A transaction that will require more than one read and write into the database is where an owner changes role or resigns and their risks have to be assigned to another person assuming that the owner owns more than one risk. (expressed in the following query below)

```
UPDATE O.staff_id
FROM Owner O, Staff S
WHERE O.staff_id = S.staff_id
AND S.staff_name = “Name”
```

Examples of Roles and Privileges of Users

- Risk and Compliance Manager, CTO, PVC, Internal Audit will have select and update privileges to the whole database.
- In addition to this the Risk and Compliance Manager and Internal Audit will have Insert and Delete privileges for the whole database.
- Risk owners and contacts will have update and select privileges to risks which they are responsible for.
- The database administrator has all the privileges of the database and only the database administrator can create and update user accounts.

A list of constrains that will be possible to ensure by the mechanisms of the Data Definition Language of RDBMS and are documented in the E-R diagram.

Entity Integrity
Referential integrity

A list of integrity constraints that would require sophisticated mechanisms to ensure integrity and are not documented in the E-R diagram.

Domain Integrity constraints are not documented in ER diagram

Critique of design

- The progress report, current controls, recommended controls in the risk table and key system contents in the system table may contain multiple values but are normally in sentences or paragraphs of texts stored as VARCHAR(max). Furthermore, there are very limited similarities between records of the identified fields so trying to express the data in first normal form is not recommended. Queries are not conducted on identified fields but on the primary keys of the identified fields instead.
- A separate table “rating” is created for storing the risk rating instead of appending the rating to the “risk” table to ensure second normal form is

achieved. Risk rating is functionally dependent upon risk impact and risk likelihood. In the case where the risk framework is updated individual records will be updated automatically with the revised framework. As seen in above in the example query (1), risk rating is one of the common queries performed in the database and performing a join to obtain the risk rating may slow down the performance. A workaround could be to create a view refreshed regularly.

- Residual Risk Rating also refers to the same framework as risk rating for determining risk. (not sure how to express this)
- Risk is not a weak entity of external reference as not all risks have a documented external reference (emails, documents). Some are provided by word of mouth.