Databases

Anna Pedersem

W6, W7, W8

1 Databases

- There are two major types of databases
  - Flat file - a type of database where all the data is contained in one file.
  - Relational database - a databse in which data is organised in a series of relationships, or two dimensional tables, where the columns (attribues) represent data fields and the rows (tuples) represent records. Linking of data between records in different files is done by means of a foreign key.
- The database system itself is made up of:
  - A database managment system (DBMS): 'the software that builds, maintains and provides access to a database. It also provides data dictionary facilities, file protection, and security against unauthorised use
  - A database: an organised collection of data items which can be accessed by a database managment system. The database may consist of several linked files.
- normalisation - is the process required during the development of a rational database. It requires an understanding of the structure of the relational database, the rules followed in the database design and the steps taken to normalise data.
- Database tables (entities) have certain properties:
  - Each entry in the table represents one data item
  - The table holds data types in columns, ie. in a given attribute all items are of the same data type such as text or integer
  - Each column or attribue has a unique name in that table
  - Duplicate rows are not allowed. Each row is identifable by a unique key.
  - Rows and colums can be viewed in any sequence at any time without affecting the table contents ie. sorted or rearranged.
- The process or normalisation is used to avoid problems in the database tables such as:
  - Data redundancy such as multiple repetitions of the same data
  - Data integrity where new data is inserted in a different form to old data or old data is only partially deleted leaving part of a record.
- focusing on development, anaylsis, design of a database
- Open system database - is where people have access to input information.

  - eg. subscription service, creating a new account, you are putting data into the databases. E-Banking, you need to create an online database
  - javascript, php, SQL

- database - is a structured collection of data to history
- purpose - to be able to process data, to anaylse, to improve the system
- **different types of data bases:**
  - centralised - all the data is in one place.
    * Adv - access all data at once,
    * Dis - if one location is compromised everything is gone, can bottleneck (too much data going in or out at once)
  - distrobuted - the data is distrobuted into numerous places, can be connected through a network
    * Adv -
    * Dis - can be hard to find where the data is
- **data warehouse** - a big building filled with servers - everything is stored in one location
  - A data warehouse is a database or collection of databases that are updated
  - This data is stored for man years
  - this database can reside on one server for a company
  - or it ca be on several severs for that company
  - The overall goal of the datawarehouse is to store data over a period over time which is used in data mining ↓
- **data mining**
  - looking for patterns, trends on the internet. Purpose is to predict future data, used for marketing. **ethical issues** - invasion of privacy. ethically incorrect to share/sell the data.
  - they get around this by you agreeing to the user licence agreement
  - is used by buisness
  - they use it to find trends which can assist sales, promotions and marketing
  - The use it to identify future planning for the company.
- **legal issues**
  - when they use data without your permission
  - hacking
- **data mart**
  - - data that has been saved in different locations, different sections of data in a buisness, a combination of these make the data warehouse
  - A data mart is a small data warehouse usually with data for just one area
  - It is still a databse on a server
  - It is queried by Data Mining software to get valuable trends in data for a company

Database managment system

- a software is used to create, maintain and use a databse in a clear and efficient way.
- when you have just a table it is called a flat database
  - all data is in 1 table, even if they are not realted to each other
  - column - field - data of the same types - attribute
  - row - record - data related to each other - entity
- tool to anaylse databases - ERD (entity relationship diagram) - because entities are related (entity = a table??)
- flat database - all the information in 1 table.
- **Normalise** - you can normalise a database by getting rid of many - to - many relationships, one 1 - M or 1 - 1
- when data is repeated (data redundancy) easy way to see that it is not normalised, if there are blank cells. When there is no primary key for an entity. When cells have not been atomised
- 3 levels of normalisation N1, N2, N3 (ideal)

ways to normailse this database

Student - <u>ID</u>, Surname

Subject - <u>CourseID</u>, Name, Teacher, Room

Report - <u>ReportID</u>, Mark

- Atomisation - making data into the simplest form it can be by breaking it up into numerous fields.
  - Address = 90 Roberts road subiaco WA 6008
  - Street number, street name, suberb, state, postcode
- Why do we normalise - prevent redundancy, update, insert, delete anaomolies.
- if an anomaly happens in a database - we can have issues.
  - **Insert Anomaly**
    - An insertion anomaly occurs when data cannot be inserted into a database due to other missing data
    - most common for fields where a foreign key must not be null, but lacks the approproate data
    - eg. A user must have a group ID as a FK however no groups have been created yet. Thus a user cannot be inserted into the database as the groupID must not be null
    - This can result in data redundancy due to the omission of data.
    -
  - **Delete Anomaly**
    - A deletion anomaly occurs when data is unintentionally lost due to the deletion of other data
    - eg. a database row contains *Username* and *User Group*, John and Fred are in the user group <u>Contributors</u>, if john and fred are removed from the database, our Contributors group will also disappear. This is because we havent normalised our data, meaning the only reference to the Contributors user group lies within the same database row (or record). Hence removing the only two references of our user group results in the loss of data accuracy and integrity.

    - This also goes to show why its important for us to normalise our data and how combining unlike information can be problematic.
- **Update Anomaly**
    - An update anomaly occurs when data is only partially updated in a database
    - A database that hasnt undergone normalisation may reference the same data element in more than one location
    - As these locations havent been consolidated and referenced, we have to make sure each location is manually updated
    - this can cause problems as we then need to spend time searching for and updating each reference to the data element
    - An example of this is a database containing two records, Users and mailing list, john has an email address in the users record and has the same address in the mailing list record. If john decides to change his email preferences, which in turn updates the User record for John, however the system did not automatically update the mailiing list record, leaving john with 2 associated emails and thus creating inconsistencies within the database.
- **Data Redundancy**
    - occurs when the same data is entered in to or more fields of a database
    - eg. Joe is entered in to the name field under a record called customers. Joe is also entered in to the customer field under a record called purchases. Although we are referring to the
- we normailse the database to get rid of these redundnacies
- **Normalise**
    - We remove many - many relationships
    - We can add entities in, group together data that is related to each other
    - we create an associate entity, can have any name, eg. Student/course, timetable
    - bc it makes an association between the two other entities.
    - associate entity always has the many entity on it and is the child entity
    - needs a primary key
- **Cardinality**
    - Represents the relationships between eneities
    - eg. 1 -1 , 1 - M, M - 1
- **attributes**
    - Primary key - underlined
    - Foreign key - (fk)

Integrities

- Referenial integrity
    - ensures that entities are related together through primary and foriegn key
    - Refential integrity states that every foreign key must reference a vlaid existing value in another table
    - this means that for every record in a normailsed database the linking element must exist in another record
    - both the primary and foreign keys myst be the same data type and length.
- Domain integrity
    - Make sure that the data inputed into the database is correct. Uses a data dictionary to make sure that each cell in a record/field is correct.

    - refers to the boundaies that shape the data entered into a database
    - This can be as simple as placing a limit on the length of the data item and enforcing a specific data type
    - Domain integrity ensures organisation and validity in a database structure
    - Data dictionary
        - contains the different fields of an entity, each field is different in terms in data types
        - must have the name of the field, data type, data format / size (can add a condition, eg. more than 10 and less than 250 characters, specify the format for the boolean, date/time format)
        - ensures that the person who enters the information (especially in an open system) enters the correct data
        - **Data types**
        - string
        - integer
        - boolean
        - date.time
        - character
        - float.float
    - metadata - descipition of data
    -
  - Entity integrity
    - Entities must have a primary key, makes sure that each entity is unquie.
    - ensures the validity of primary keys
    - the concept states that each primary key must not be NULL
    - it also states that each primary key must be unquie, meaning no pk value may be the same as another pk in the same record.

ERDs

  - ID is not underlined, foreign key does not have (fk), Recommended and price
  -

SQL Queries

SQL - structured query language

- SELECT - what fields you want to include (required), most common function. * represents everything.
  - SELECT DISTINCT - only list the distinct values
- FROM - what databases you want to get fields from (required)
- WHERE - extract records that fulfill a specific condition
- ORDER BY - specifies how to sort the results
- GROUP BY - In SQL statement that contains aggregate functions, list fields that are not summarised in the SELECT cause
- HAVING - in a SQL statement that contains aggregate functions, specifices conditions that apply to fields that are summarised in the SELECT cause
- UPDATE - Update data in a database

- DELETE - used to delete data from a database
- INSERT INTO - insert new data in a database
- INNER JOIN - most common type of join
- BETWEEN - select values within a range
- CREATE TABLE - create a database

how do you select all the records from a table named "Persons" where "lastName" is alphabetically between (and including) "Hansen" and "Pettersen"

```
SELECT * FROM Persons WHERE LastName BETWEEN 'Hansen' AND 'Pettersen'
```

How can you return all the records from a table named "Persons" sorted desending by "firstName"

```
SELECT * FROM Persons ORDER BY FirstName DESC
```

2 Open systems in databse interconnectivity

- Role for open systems in database
- Types of SQL
  - DML (Data manipulation language) - DML enables you to work with the data that goes into the database. DML is used to isert, select, update and delete records in a database dealing with the manipulation of database. Many of your SQL statments will begin with one of the following commands:
    - SELECT - retrieves data from the database
    - INSERT - Inserts new data into the database

      ```
      INSERT INTO 'example'.'teachers' ('idteachers', 'TeacherFirstName',
      'TeacherTitle') VALUES ('1', 'Tom', 'Stokes', 'Mr');
      ```

    - UPDATE - Updates existing data in the database, change Hansen into Nilsen in the LastName column of the persons table

      ```
      UPDATE Persons SET LastName='Nilsen' WHERE LastName='Hansen';
      ```

    - DELETE - Deletes existing data from the database

      ```
      DELETE FROM `example`.`students` WHERE `idstudents` = 1;
      ```

    - LIKE - value FirstName starts with a

      ```
      SELECT * FROM Persons WHERE FirstName LIKE 'a%';
      ```

  - DDL (Data definition language) - You may also occasionally need to create or drop a table or other database objects SQL ebables you to do this programmically using DDL.
    - CREATE DATABASE - Creates a new database
    - ALTER DATABASE - Modifies the database
    - DROP DATABASE - Drops (deletes) a database
    - CREATE TABLE - Creates a new table

```
CREATE TABLE 'example'.'teachers' (
  'teacherID' INT NOT NULL,
  'teacherFirstName' VARCHAR(45) NULL,
  'teacherLastName' VARCHAR(45) NOT NULL,
  PRIMARY KEY('teacherID'));

CREATE TABLE `example`.`hardware` (
`hardwareID` INT NOT NULL,
`productName` VARCHAR(45) NOT NULL,
  `idPeople` INT NOT NULL,
  PRIMARY KEY(`hardwareID`),
  FOREIGN KEY(`idPeople`) REFERENCES `example`.`people` (`idPeople`) );
```

* ALTER TABLE - Modifies the table

```
ALTER TABLE `example`.`subject` RENAME TO  `example`.`subjects`;

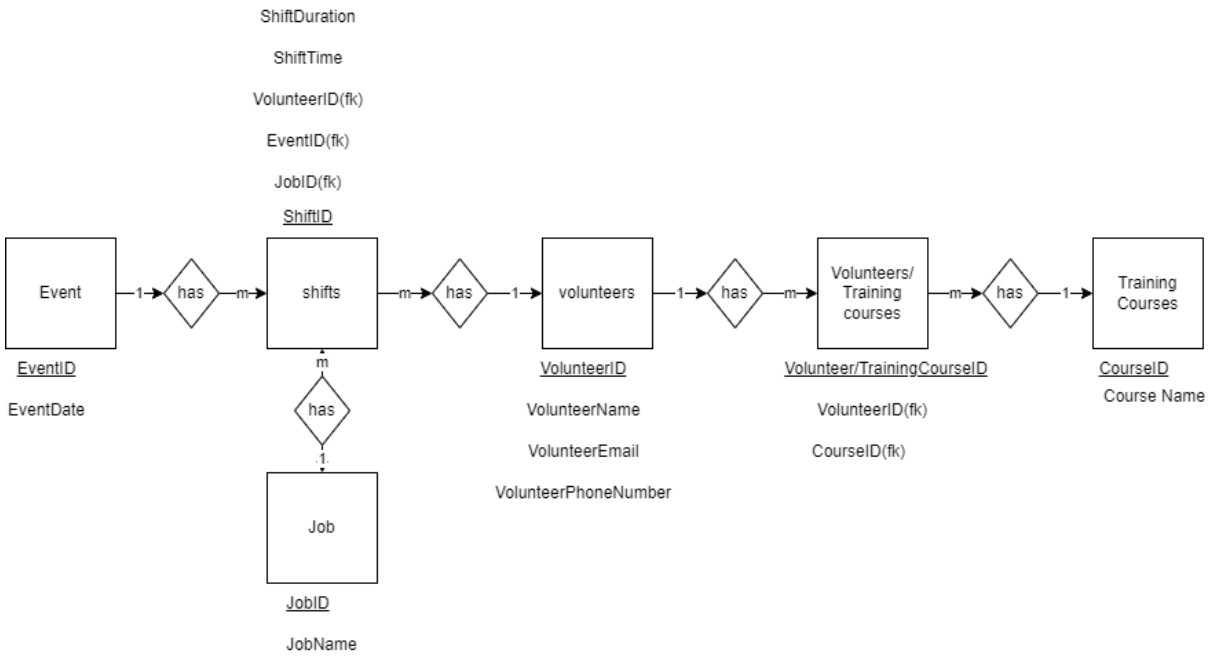
ALTER TABLE `example`.`subjects` ADD COLUMN `idteachers`
INT NOT NULL AFTER `SubjectName`;
```

* DROP TABLE - Drops (deletes) a table
* ADD CONSTRAINT

- Wild cards
  - . to match any single character
  - [] to match any character in the bracket [xyz], will match with "x","y","z"
  - Same can be used to match a character range seperated by - [a-z] will mtach any characters between "a", and "z"
  - * matches any number of characters preceding it
  - m̂atch the beginning of the string
  - $ match the end of value
  - % Select entries that contain something
    - ```
      SELECT * FROM people WHERE firstName LIKE "a%";
      ```
    - returns only the people entries who start with a
- Role for open systems in database interconnectivity and the development and management in data driven websites
- **Background**
  - In order to link to a website the provides a database it needs a system of connectivity
  - This connectivity is a special program called an Application Program Interface (API). API is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, you send an instant message, or check the weather you're using an API
- **Database Connectivity**
      ```

- Picture 2 components here; you on the internet and a database on the computer
- To get to that database it needs to be made avaliable on the internet
- It can only be made avaliable on the internet by using a speical type of software called an API
- It can also be made avaliable using Open Database Connectivity (ODBC) software. ODBC is an opern standard applucation programming interface (API) for accessing a database
- The API or ODBC made the connection of database to the internet happen.

3 Class practice questions



4 Practice WACE Questions

1. Data integrity is a database can be divided into 3 catagories: referential integrity, domain integrity and entity integrity

   (a) Outline the meaning of each of the following (2 marks)

   - Referenial integrity → Referenial integrity is ensures that entities are related to each other through primary and foreign keys, it states that a foreign key must be a valid, existing value in another table
   - Two entities that are related require that a foreign key musy have a matching primary key
   - Entity integrity → Entity integrity means that each entity must have a valid primary key
   - Entity integrity specificies that the primary keys on every instance of an entity must be kept, must be unique and must have values other than NULL

   (b) Describe how data integrity can improve the process of database managment (2 marks)

   - Data integrity is the maintence of, and the assurance of, data accuracy and consistency over its entire life-cycle and is a critical aspect to the design, implementation and usage of any system that stores, processes or retrieves data. The overall intent of any data integrity is the same: ensure data is recorded excatly as intend, such as a database correctly rejecting mutually exclusive possibilities.

5 2021 Exam

-