

Limitation of a file-based approach

Data integrity problems in a single file:

Database approach can prevent the lack of in-built control when data is entered.

The data privacy issue with a single file:

The problem is that there cannot be any control of access to part of a file. Data privacy would be properly handled by a database system.

Data redundancy and possible inconsistency in multiple files:

One of the primary aims of the database approach is the elimination of data redundancy.

Data dependency concerns:

In a database scenario the existing programs could still be run even though additional data was added.

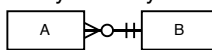
Entity-relationship modelling

one-to-one 1:1

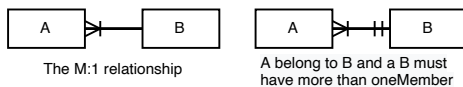
one-to-many 1:M

many-to-one M:1

many-to-many M:M

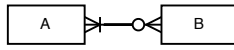


One A is for one B (There must be a B and there cannot be more than one) and one B can be used for many A

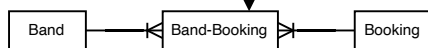


The M:1 relationship

A belong to B and a B must have more than oneMember



M:M relation



M:M relation

The Database Management System (DBMS)

Three-level model:

1. **The external level:** The individual's view(s) of the database

2. **The conceptual level:** Describes the data as seen by the applications making use of the DBMS. Describes the 'views' which users of the database might have

3. **the internal level:** Describes how the data will be stored on the physical media

Data management system (DBMS): software that controls access to data in a database

Database administrator (DBA): a person who uses the DBMS to customise the database to suit user and programmer requirements

An important aspect of the provision of views is that they can be used by the DBA as a mechanism for ensuring security. Individual users or groups of users can be given appropriate access rights to control what actions are allowed for that view.

Developer interface: gives access to software tools provided by a DBMS for creating tables

task:

1. create table
2. set up relationship between tables
3. create / design a form
4. create / design a report
5. create / design a query

Query processor: software tools provided by a DBMS to allow creation and execution of a query

Query: used to select data from a database subject to defined conditions

Data dictionary: contains metadata about the data. This includes details of all the definitions of tables, attributes and so on but also of how the physical storage is organised.

1. table name
2. field name // attribute
3. data type
4. type of validation
5. Primary Key
6. Foreign Key
7. relationships

index: a small secondary table used for rapid searching which contains one attribute from the table being searched and pointers to the tuples in that table

Database Security:

1. Issue usernames and passwords
2. Access rights
3. Create backups
4. Encryption of data
5. Definition of different views
6. Usage monitoring

The relational database

Relation: the special type of table which is used in a relational database

Attribute: a column in a relation that contains values

Tuple: a row in a relation storing data for one instance of the relation

Primary key: an attribute or a combination of attributes for which there is a value in each tuple and that value is unique

Candidate key: a key that could be chosen as the primary key

Secondary key: a candidate key that has not been chosen as the primary key

Foreign key: an attribute in one table that refers to the primary key in another table

Referential integrity: the use of a foreign key to ensure that a value can only be entered in one table when the same value already exists in the referenced table

Normalisation

1NF:

1. Ask if there is repeating group attributes.
2. For each table that the attributes are dependent on the primary key

2NF:

1. Examine each non-key attribute and ask if it is dependent on both parts of the component key.
2. it either has a single primary key or it has a compound primary key with any non-key attribute dependent on both component.

3NF:

1. see if there any non-key dependencies
2. look for any non-key attribute that is dependent on another non-key attribute

Repeating group: a set of attributes that have more than one set of values when the other attributes each have a single value

Structured Query Language (SQL)

Data Definition Language: provide for creating or alter tables.

CREATE DATABASE BandBooking;

CREATE TABLE Band (

BandName varchar(25),

NumberOfMembers integer);

ALTER TABLE Band ADD PRIMARY KEY (BandName);

ALTER TABLE Band-Booking ADD FOREIGN KEY (BandName

REFERENCES Band(BandName);

ALTER TABLE Band ADD Gender CHAR(1); // add an attribute

Data type : character, varchar, boolean, integer, real, date, time

Data Manipulation language(DML):

1. The insertion of data into the tables when the database is created
2. The modification or removal of data in the database
3. The reading of data stored in the database

Insert:

INSERT INTO Band ('ComputerKidz', 5);

INSERT INTO Band-Booking (BandName, BookingID) VALUES ('ComputerKidz', '2016/023');

Select:

SELECT BandName FROM Band;

SELECT BandName, NumberOfMembers FROM Band;

SELECT * FROM Band;

Order and Group:

SELECT BandName, NumberOfMembers FROM Band ORDER BY BandName;

SELECT BandName FROM Band-Booking GROUP BY BandName;

SELECT BandName FROM Band-Booking WHERE Headlining = 'Y' GROUP BY BandName;

SELECT BandName, NumberOfMembers FROM Band WHERE NumberOfMembers > 2 ORDER BY BandName;

Aggregate functions (SUM,COUNT,AVG):

SELECT Count(*) FROM Band;

SELECT AVG(NumberOfMembers) FROM Band;

SELECT SUM(NumberOfMembers) FROM Band;

Join condition:

SELECT VenueName, Date FROM Booking WHERE Band-Booking.BookingID = Booking.BookingID AND Band-Booking.BandName = 'ComputerKidz';

INNER JOIN:

SELECT table1.column1, table2.column2... FROM table1 INNER JOIN table2 ON table1.common _ field = table2.common _ field;

DML UPDATE:

UPDATE Band SET NumberOfMembers = 6 WHERE BandName = 'ComputerKidz';

DML Delete:

DELETE FROM Band-Booking WHERE BandName = 'ITWizz'; DELETE FROM Band WHERE BandName = 'ITWizz'