
DATABASES

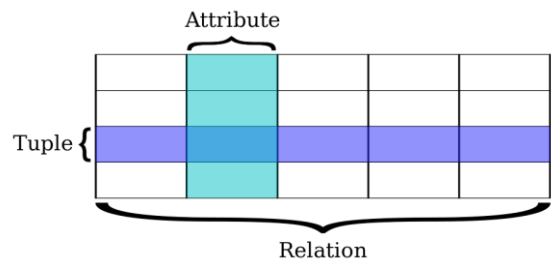
Relational Databases

Method of creating a database using tables of related data, with relationships between tables



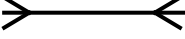
✚ Table: a method for implementing one entity and attributes as a group of related data

✚ Entity: an object about which data will be stored

✚ Attribute: a characteristic or piece of information about an entity, which would be stored as a field in a relational database



Entity relationship diagram: describes relationships between entities. Two entities can be linked by a relation:

- One-to-One 
- One-to-Many 
- Many-to-Many 

✚ Primary key: an attribute that can be used to uniquely identify every record in a table. It is composite if two or more attributes are used in combination.

✚ Foreign key: an attribute in a table that is a primary key in another table and is used to link tables together.

Standard notation:

Patient (PatientId, Name,
Address, DoB, WardNo, StaffNo)

- Entity name outside the brackets
- Attributes inside the brackets
- Primary key (Entity identifier) underlined
- Foreign keys in italics

Normalisation

Process of ensuring that a relational database is structured efficiently. In particular a normalised database must have:

- *No redundant data*: same field unnecessarily duplicated in two or more tables → so less space required to store the data and no inconsistency problems
- *Data stored at atomic level*: they cannot be further decomposed. For example no fields Address containing the full address of the customer, divide in HouseNumber, Street, County...

There are various level of normal form:

1. FIRST NORMAL FORM:

A record is problematic if: it contains repeating groups of attributes (a group of values is stored in a particular row/column intersection) or it contains non-atomic data.

Groups of attributes are spitted creating one record for each.

Student No	Student Name	DoB	Course No	Course Name	Lecturer No	Lecturer Name
12345	Heathcote, R	20/08/1983	EC6654	A Level Computing	T345267	Glover, T
22433	Head, J	13/02/1983	EC6654	A Level Computing	T345267	Glover, T
			HM7756	A Level Music	T773351	Reader, B
			AD1121	Pottery	T876541	Day, S
66688	Hargrave, R	13/09/1954	BM3390	HNC Business	T666758	Newman, P
			HM7756	A Level Music	T773351	Reader, B

2. SECOND NORMAL FORM:

A record is going to be problematic if: it has a composite primary key and an attribute is dependent on only part of the composite key. It is solved by creating a new table which contains the composite key and connect the new table to originals by relationships.

Student No	Course No	Student Name	DoB	Course No	Course Name	Lecturer No	Lecturer Name
12345	EC6654	Heathcote, R	20/08/1983	EC6654	A Level Computing	T345267	Glover, T
22433	EC6654	Head, J	13/02/1983	HM7756	A Level Music	T773351	Reader, B
22433	HM7756	Head, J	13/02/1983	AD1121	Pottery	T876541	Day, S
22433	AD1121	Head, J	13/02/1983	BM3390	HNC Business	T666758	Newman, P
66688	BM3390	Hargrave, R	13/09/1954				
66688	HM7756	Hargrave, R	13/09/1954				

3. THIRD NORMAL FORM:

A record is going to be problematic if: one or more attributes are not dependent on the primary key. It is solved by: creating a new table which contains the attribute(s) and connecting the new table to originals by relationships.

Student No	Student Name	DoB	Student No	Course No	Course No	Course Name	Lecturer No	Lecturer Name
12345	Heathcote, R	20/08/1983	12345	EC6654	EC6654	A Level Computing	T345267	Glover, T
22433	Head, J	13/02/1983	22433	EC6654	HM7756	A Level Music	T773351	Reader, B
66688	Hargrave, R	13/09/1954	22433	HM7756	AD1121	Pottery	T876541	Day, S
			22433	AD1121	BM3390	HNC Business	T666756	Newman, P
			66688	BM3390				
			66688	HM7756				

Normalisation process summary

0. Write out all the fields in 1 table 1st
 - a. Separate out into obvious tables next
1. 1st Normal Form
 - a. Look for repeating data
 - b. Make the repeating data part of the key
2. 2nd Normal Form
 - a. Look for composite keys
 - b. Are there any fields dependent on only part of the key?
 - c. Create a new table containing the composite key
3. 3rd Normal Form
 - a. Are there any fields not dependent on the key?
 - b. Create a new table containing those fields

SQL

Structured query language SQL: a specialised programming language for manipulating databases

Define a table:

```
CREATE TABLE Employee
(EmpId INT PRIMARY KEY,
Name VARCHAR(10),
HiredDate DATE,
DepId INT,
FOREIGN KEY (DepId) REFERENCES Dep ( DepId ));
```

if composite key at the end: PRIMARY KEY (EmpId, CourseDate)

Entering data:

```
INSERT INTO Dep (DepId, DepName, Extention)
VALUES (1, "Technical", "1340");
```

Update data:

```
UPDATE Employee
SET Salary = Salary *1.1
WHERE DepId = 1;
```

Delete data:

```
DELETE FROM Employee
WHERE DepId = 1;
```


Querying data:

```
SELECT * FROM Employee
WHERE EmpName = "John Smith"
ORDER BY EmpName DESC;
```

```
SELECT CustomerName, Address, DateOfDownload, Movie.MovieID, Price
FROM Customer, Download, Movie
WHERE Download.DateOfDownload = "19/03/15" AND
Customer.CustomerID = Download.CustomerID AND
Movie.MovieID = Download.MovieID
```

Client-server databases

Client-server database is a way of implementing a database where the database is put into a server and various user can access it from their workstations. The processing, for example running a query, will take place on the server.

 Database management system (DBMS): software that enables the management of all aspects of a database including adding, updating and querying data

One common problem with such a database is concurrent access: what to do if a number of users are trying to write data in the same location at the same time.

- *Record locks*: a technique to temporarily prevent access to certain records held on a database
- *Serialisation*: a technique to ensure that only one transaction at the time is executed from multiple users on a database
- *Timestamp ordering*: a technique to ensure multiple users can execute commands on a shared database based on the timestamp of when the data was last written to or read from
- *Commitment ordering*: a technique to ensure concurrent transaction on a shared database are executed based on the timestamp of when the request is made and also the precedence the request takes over other simultaneous requests