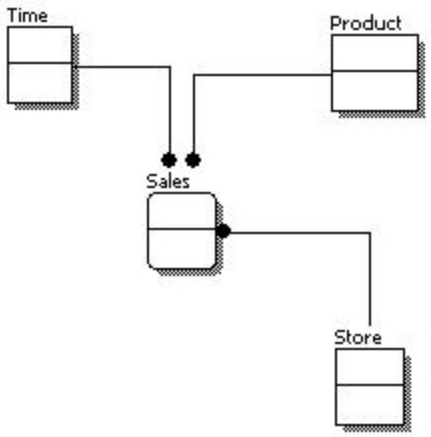
Topic
Database Normalisation

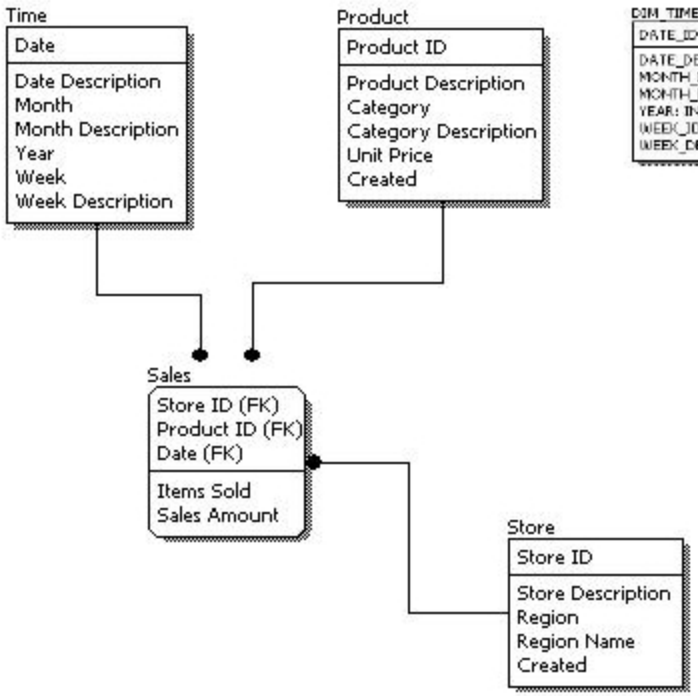
3 phases…

Need a systematic approach to designing and building the dB

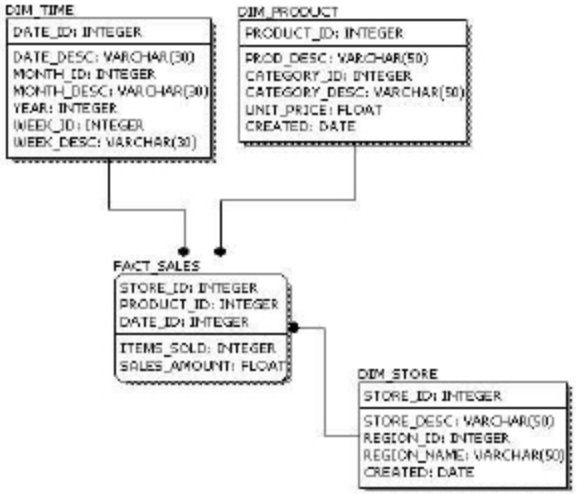
Conceptual



Logical



Physical



- **Normalisation** is a process by which data structures in a relational database are as efficient as possible, including the elimination of redundancy, the minimisation of the use of null values and the prevention of the loss of information.

Aims of Normalisation

- Normalisation ensures that the database is structured in the best possible way.
- To achieve control over data redundancy. There should be no unnecessary duplication of data in different tables.
- To ensure data consistency. Where duplication is necessary the data is the same.

- To ensure tables have a flexible structure.
  - (e.g. number of order lines per order not limited)
- To make the database suitable for querying

Duplication vs Redundant Data

- Duplicated Data: When an attribute has two or more identical values
- Redundant Data: If you can delete data without a loss of information

What's wrong with this?

Number	Name	Classes
001231	William Hartnell	Information Systems, Systems Analysis, Data Communications
001232	Patrick Troughton	Systems Analysis, Data Communications
001233	Jon Pertwee	OO Programming, Systems Analysis, Data Communications
001234	Tom Baker	Systems Analysis, Data Communications

Student table

And this?

Number	Name	Classes
001231	William Hartnell	Information Systems
001231	William Hartnell	Systems Analysis
001231	William Hartnell	Data Communications
001232	Patrick Troughton	Systems Analysis
001232	Patrick Troughton	Data Communications
001233	Jon Pertwee	OO Programming
001233	Jon Pertwee	Systems Analysis
001233	Jon Pertwee	Data Communications
001234	Tom Baker	Systems Analysis
001234	Tom Baker	Data Communications

What about this?

RefNo	Name	Address	Status	AccNo
345	C.J. Date	23, High Street	Business	120768
345	C.J. Date	23, High Street	Business	348973
543	F.D. Rolland	45, The Ash	Domestic	987654
675	D.R. Howe	17, Low Street	Business	745363
675	D.R. Howe	17, Low Street	Business	678453
675	D.R. Howe	17, Low Street	Business	348973

Is this ok?

Branch Name	Address	Manager No	Acc No	Balance	Type
Rathmines	15 Upr Rathmimes Road	1234	1205	-£123.45	‘D’
Rathmines	15 Upr Rathmimes Road	1234	6784	£67.54	‘C’
Rathmines	15 Upr Rathmimes Road	1234	9843	£43.43	‘C’
Dame St.	1 Dame Street	1101	5422	£34.50	‘C’
Dame St.	1 Dame Street	1101	0998	£666.66	‘D’

Is this ok?

Restaurant	Pizza Variety	Delivery Area
A1 Pizza	Thick Crust	Springfield
A1 Pizza	Thick Crust	Shelbyville
A1 Pizza	Thick Crust	Capital City
A1 Pizza	Stuffed Crust	Springfield
A1 Pizza	Stuffed Crust	Shelbyville
A1 Pizza	Stuffed Crust	Capital City
Elite Pizza	Thin Crust	Capital City
Elite Pizza	Stuffed Crust	Capital City
Vincenzo's Pizza	Thick Crust	Springfield
Vincenzo's Pizza	Thick Crust	Shelbyville
Vincenzo's Pizza	Thin Crust	Springfield
Vincenzo's Pizza	Thin Crust	Shelbyville

There are increasing levels of normalisation

- 1st normal form – 1NF
- 2nd normal form – 2NF
- ..
- 4th normal form
- 5th normal form

- They're cumulative – e.g. a database can't be in 4th normal form, without being in 1st, 2nd, 3rd also

In the real world, normalisation of databases follow a set of design steps

Candidate Key

- A *Candidate Key* is an attribute (possibly composite) that can be used to uniquely identify each tuple in a relation.
- A relation may have more than one candidate key
- If so, one candidate key is nominated as the primary key

First Normal Form

A table is in its first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain.

Non-Normalised Table

STUDENT

Number	Name	Classes
001231	William Hartnell	Information Systems, Systems Analysis, Data Communications
001232	Patrick Troughton	Systems Analysis, Data Communications
001233	Jon Pertwee	OO Programming, Systems Analysis, Data Communications
001234	Tom Baker	Systems Analysis, Data Communications

First Normal Form

INCONSISTENT

Number	Name	Classes
001231	William Hartnell	Information Systems
001231	William Hartnell	Systems Analysis
001231	William Hartnell	Data Communications
001232	Patrick Troughton	Systems Analysis
001232	Patrick Troughton	Data Communications
001233	Jon Pertwee	OO Programming
001233	Jon Pertwee	Systems Analysis
001233	Jon Pertwee	Data Communications
001234	Tom Baker	Systems Analysis
001234	Tom Baker	Data Communications

Second Normal Form

A table is in the second normal form if it's in the first normal form AND non-prime attributes are functionally dependent on the whole of every candidate key.

(the values in each non primary key column can be worked out from the values in *all* the candidate key(s) i.e. you can‟t work out the values from part of the candidate key)

Is this in Second Normal Form

The values in each **non primary key column** can be worked out from the values in *all* the candidate key(s) i.e. you can‟t work out the values from part of the **candidate key**

Employee	Skill	Office
Murphy	Typing	22 Barn Way
Murphy	Cooking	22 Barn way
Murphy	Cleaning	22 Barn way
Keane	Basic cooking	114 Rodeo drive
Maguire	Programming	9 Harbour view
Maguire	Typing	9 Harvour view
Doyle	Hardware config	10 The close

What‟s the candidate key? What are the non primary key columns?

Second Normal Form

- The concept of functional dependency in central to normalisation and, in particular, strongly related to 2NF.

- Think of the word functional dependant as “can be worked out from”

Functional Dependency

- If „X“ is a set of attributes within a relation, then we say „A“ (an attribute or set of attributes), is functionally dependant on X, if and only if, for every combination of X, there is only one corresponding value of A
- We write this as :

X -> A

Table in 1NF

RefNo	Name	Address	Status	AccNo
345	C.J. Date	23, High Street	Business	120768
345	C.J. Date	23, High Street	Business	348973
543	F.D. Rolland	45, The Ash	Domestic	987654
675	D.R. Howe	17, Low Street	Business	745363
675	D.R. Howe	17, Low Street	Business	678453
675	D.R. Howe	17, Low Street	Business	348973

- Remember what‟s wrong with it?
- Can you work out the name, address, status from the primary key or from something else?

Functional Dependency

- It is clear that :

RefNo -> Name, Address, Status

i.e.

Name, address and status are functionally dependant on Ref No

OR

Name address status can be worked out from Ref no…

Because for every ref no there is only one name, address, status)

or, most correctly,

AccNo, RefNo -> Name, Address, Status

Second Normal Form

RefNo	AccNo
345	120768
345	348973
543	987654
675	745363
675	678453
675	348973

RefNo	Name	Address	Status
345	C.J. Date	23, High Street	Business
543	F.D. Rolland	45, The Ash	Domestic
675	D.R. Howe	17, Low Street	Business

Over to you…

Supplier#	Part#	City	Quantity
S1	P1	London	1000
S1	P2	London	1500
S1	P3	London	3400
S1	P4	London	2100
S2	P2	Paris	3400
S2	P3	Paris	1000
S4	P1	Nuku alofa	5
S4	P4	Nuku alofa	7

Table in Second Normal Form

Supplier#	City
S1	London
S2	Paris
S4	Nuku alofa

Supplier#	Part#	Quantity
S1	P1	1000
S1	P2	1500
S1	P3	3400
S1	P4	2100
S2	P2	3400
S2	P3	1000
S4	P1	5
S4	P4	7

Third Normal Form

A table is in the third normal form if it is the second normal form and there are no *non-key columns* dependant on *other non-key columns* that could not act as the primary key.

Table in Second Normal Form

Branch Name	Address	Manager No	Acc No	Balance	Type
Rathmines	15 Upr Rathmimes Road	1234	1205	-£123.45	'D'
Rathmines	15 Upr Rathmimes Road	1234	6784	£67.54	'C'
Rathmines	15 Upr Rathmimes Road	1234	9843	£43.43	'C'
Dame St.	1 Dame Street	1101	5422	£34.50	'C'
Dame St.	1 Dame Street	1101	0998	£666.66	'D'

Are there any *non-key columns* dependant on *other non-key columns* (that could not act as the primary key.)

Table in Third Normal Form

Acc No	Balance	Type	Branch Name
1205	-£123.45	‘D’	Rathmines
6784	£67.54	‘C’	Rathmines
9843	£43.43	‘C’	Rathmines
5422	£34.50	‘C’	Dame St.
0998	£666.66	‘D’	Dame St.

Branch Name	Address	Manager No
Rathmines	15 Upr Rathmimes Road	1234
Dame St.	1 Dame Street	1101

Important Papers

- E.F.Codd „*A Relational Model for Large Shared Data Banks'* CACM 13(6) June 1970
- E.F. Codd „*Extending the Database Relational Model to Capture More Meaning'* ACM Transactions on Database Systems, 4(4), December 1979

Table in Third Normal Form

CourseNo	Lecturer	Time	Room
FT225/2	P. O'Byrne	9.00	121
FT222/1	D. Gordon	9.00	666
DT266/2	K. O'Brien	1.00	343
DT266/2	D. Carroll	11.00	876
FT222/4	K. O'Brien	3.00	343
FT228/3	D. Gordon	3.00	666

Redundancy in 3NF

- The combination of ROOM, TIME is unique to each tuple, no room is used twice at the same time -> key
- But, we know there is a redundancy in that ROOM depends LECTURER, therefore, we split the table...

Fourth Normal Form

The 4NF removes unwanted data structures: multivalued dependencies.

Fourth Normal Form

Consider employees, skills, and languages, where an employee may have several skills and several languages. We have here two many-to-many relationships, one between employees and skills, and one between employees and languages. Under fourth normal form, these two relationships should not be represented in a single record such as

| EMPLOYEE | SKILL | LANGUAGE |

Instead, they should be represented in the two records

| EMPLOYEE | SKILL |

| EMPLOYEE | LANGUAGE

In practice, do the following checks as you design, before “thinking” about normalisation

- Group data into tables, with columns (obviously)
- Identify a primary key for each table
  - Might be more than one field..
- Take out any repeating groups and put them on a separate table
  - E.g. empID, name, project1, time1, project2, time2, project3,time3, etc
- Make sure every **non-key** column depends on the primary key.
  - e..g on an Orders table, columns such as Order_date and Order_amount are valid. Storing client's details such as name, address and so on are not valid and should be on a separate table

In practice, do the following checks as you design go, before “thinking” about normalisation (2)

- Remove repeating columns to a new table. Repeating columns are indicative of a missing relationship.

e.g. on an Orders table it will not have item1, item2, and item3 columns for all items purchased. The items will be removed to a *separate* table linked to the orders table.

Remove repetition within columns.

e.g. a company may accept Visa, Mastercard and American Express for payment. Rather than repeat those strings in the order payment table, they can be places in a small lookup table and the order payment table can refer to their smaller primary key.

i.e. orderID, date, payment type payment code, card description

08989 1/2/10 v v Visa