



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022

Database Architecture

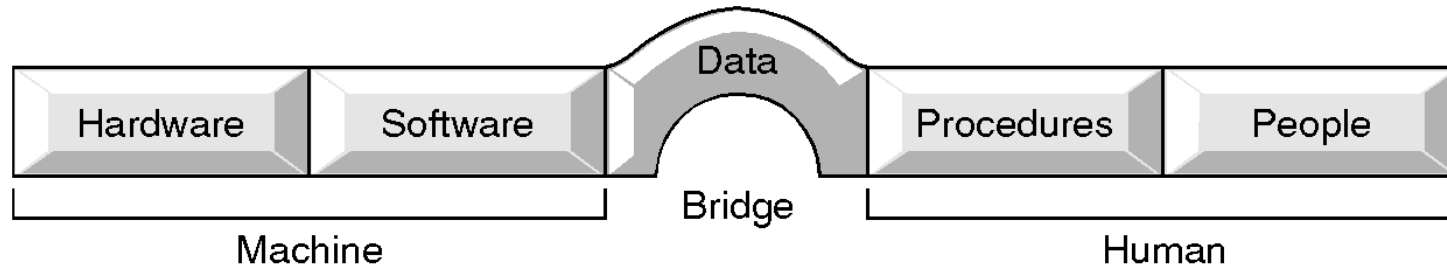
Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

www.uwtsd.ac.uk

Objectives

- Data modelling concept
- Types of data modelling techniques
- Conceptual data modelling
- Logical data modelling
- Physical data modelling
- Conclusion

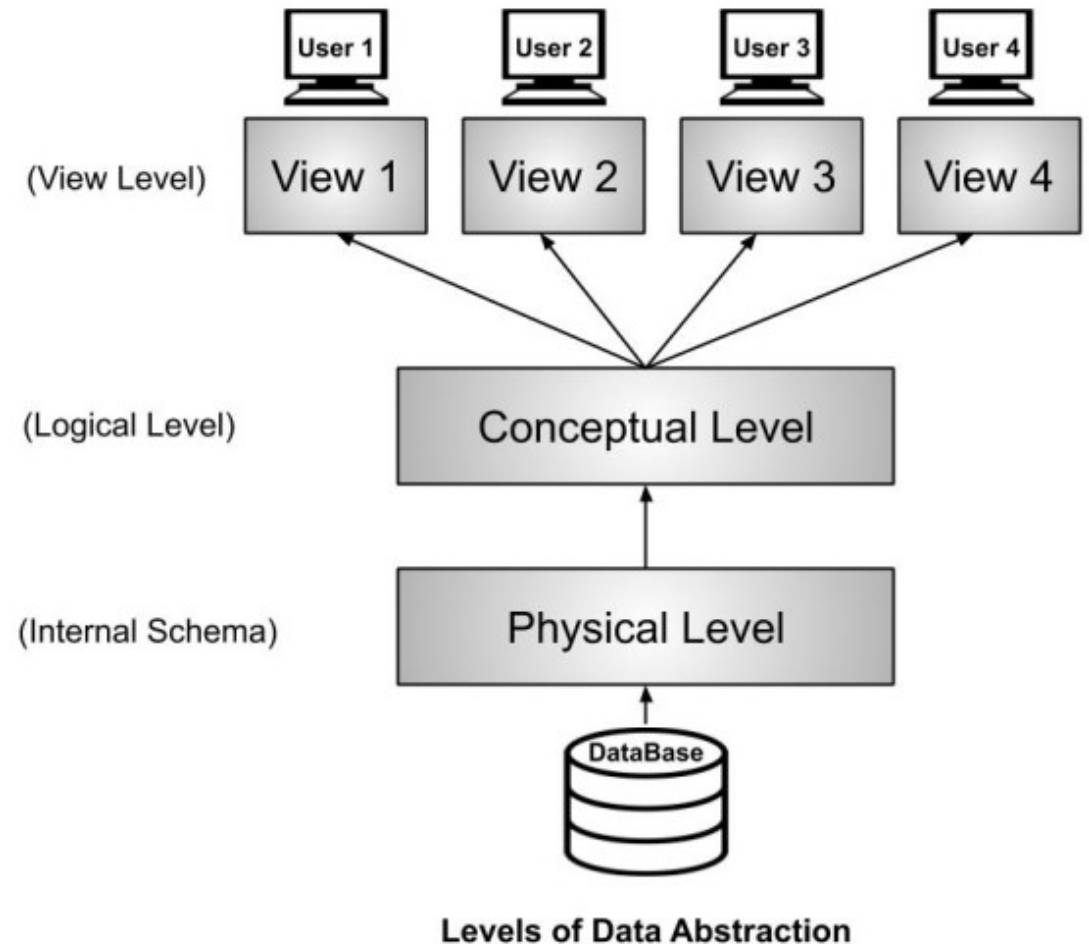
Components of DBMS Environment



- **Hardware**
 - Can range from a mobile phone to a laptop to a network of computers.
- **Software**
 - DBMS, operating system, network software (if necessary) and also the application programs.
- **Data**
 - Used by the organization and a description of this data called the schema.
- **Procedures**
 - Instructions and rules that should be applied to the design and use of the database and DBMS.
- **People**
 - The individuals who interact with the system

Levels of abstraction(Architecture)

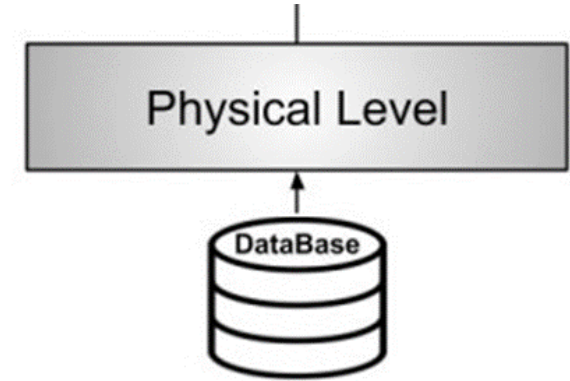
- Three levels of abstraction
- Physical or Internal Level
- Logical or Conceptual Level
- View or External Level



Physical or Internal Level

- Defines
 - how the data is actually stored
 - data-structures to store data and access methods
- For example, customer's information is stored in tables and data is stored in the form of blocks of storage such as bytes, gigabytes etc.

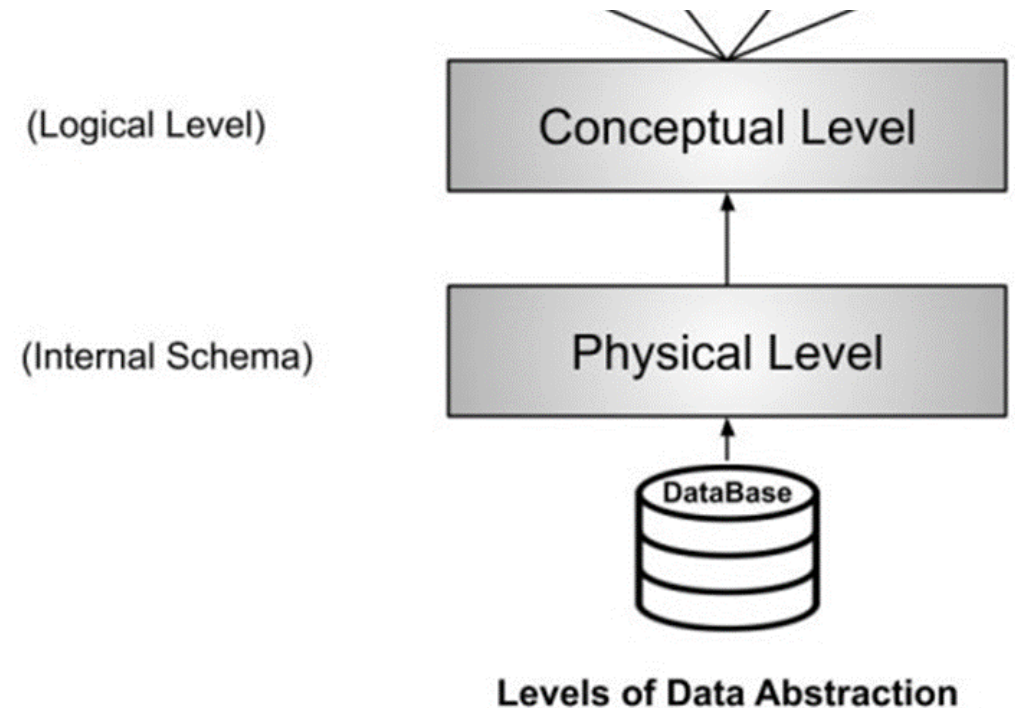
(Internal Schema)



Levels of Data Abstraction

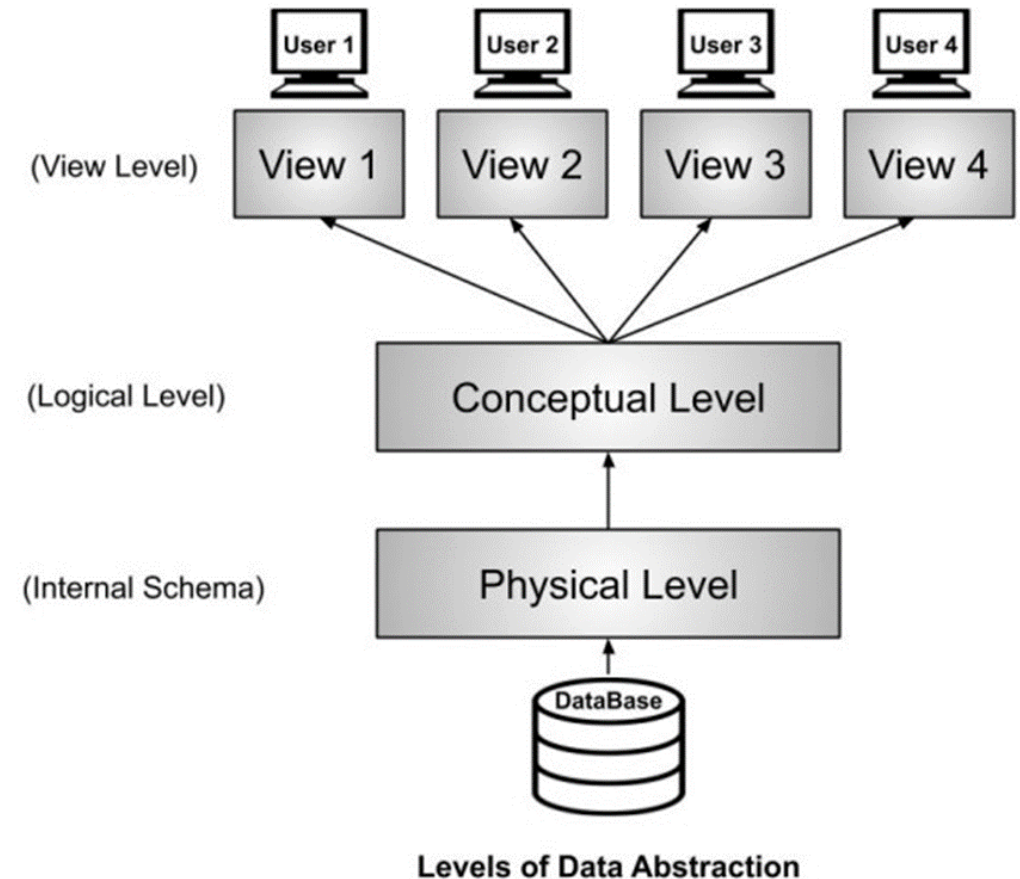
Logical or Conceptual Level

- Describes
 - what data is stored in the database
 - what relationship exists among those data
- Database designers and developers who define the structure of the database, such as tables, relationships, and constraints.



View or External Level

- The users' view of the database.
- describes the part of the database that is relevant to each user.
- number of different external views of the database



External view 1

sNo	fName	lName	age	salary
-----	-------	-------	-----	--------

External view 2

staffNo	lName	branchNo
---------	-------	----------

Conceptual level

staffNo	fName	lName	DOB	salary	branchNo
---------	-------	-------	-----	--------	----------

Internal level

```
struct STAFF {  
    int staffNo;  
    int branchNo;  
    char fName [15];  
    char lName [15];  
    struct date dateOfBirth;  
    float salary;  
    struct STAFF *next;  
};  
index staffNo; index branchNo;
```

/* pointer to next Staff record */
/* define indexes for staff */

Figure 2.2 Differences between the three levels.

External view 1

sNo	fName	lName	age	salary
-----	-------	-------	-----	--------

External view 2

staffNo	lName	branchNo
---------	-------	----------

Conceptual level

staffNo	fName	lName	DOB	salary	branchNo
---------	-------	-------	-----	--------	----------

Internal level

```
struct STAFF {  
    int staffNo;  
    int branchNo;  
    char fName [15];  
    char lName [15];  
    struct date dateOfBirth;  
    float salary;  
    struct STAFF *next;  
};  
index staffNo; index branchNo;
```

/* pointer to next Staff record */
/* define indexes for staff */

Figure 2.2 Differences between the three levels.

- **Summary of Levels of Abstraction:**

1.Physical Level: How data is stored (internal structure).

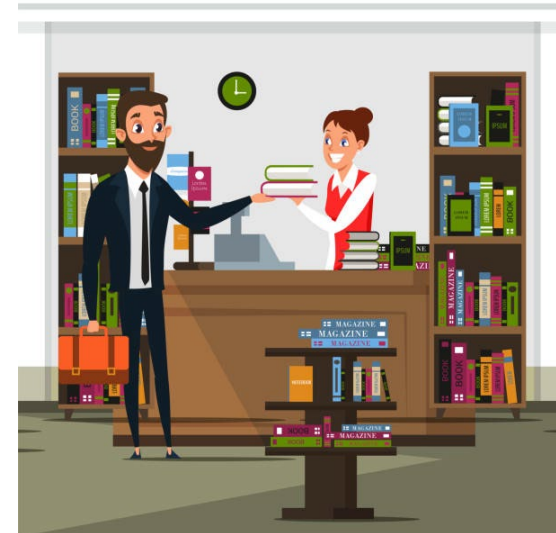
2.Logical Level: What data is stored (schema design).

3.View Level: How data is presented to users (user-specific views).

DBMS-Three Level Architecture



This level represents the view of the database from the perspective of individual users or specific applications.



DBMS-Three Level Architecture

External Layout View
End-Users



View 1

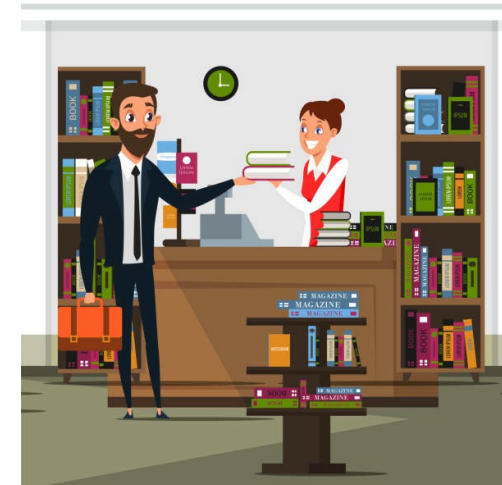


View 2

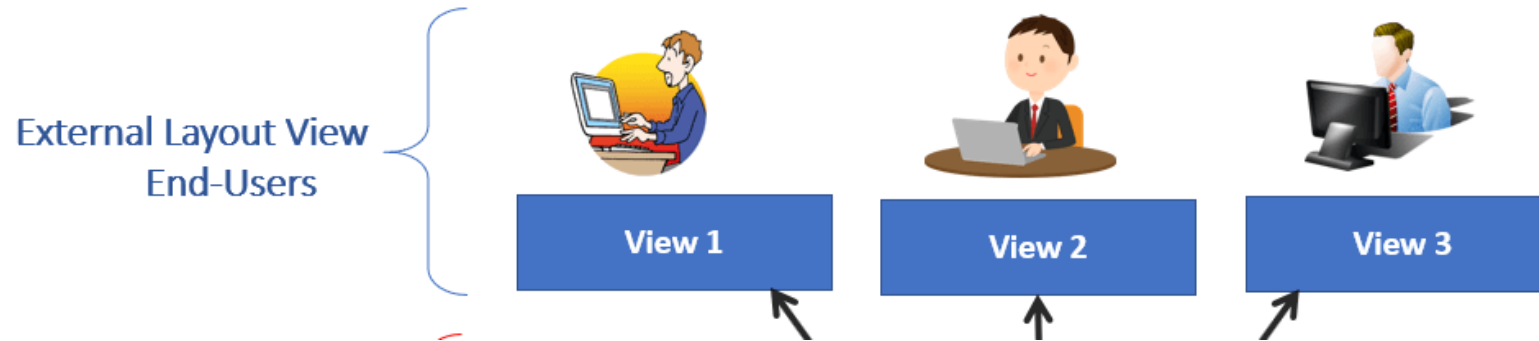


View 3

Who are the users of the book store?



DBMS-Three Level Architecture



- Customers

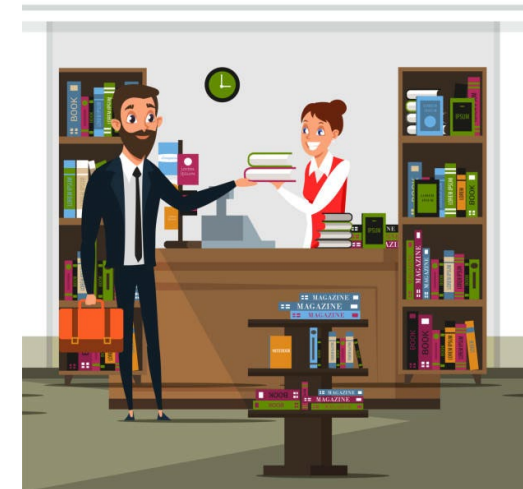
- Cashiers:

- Managers:

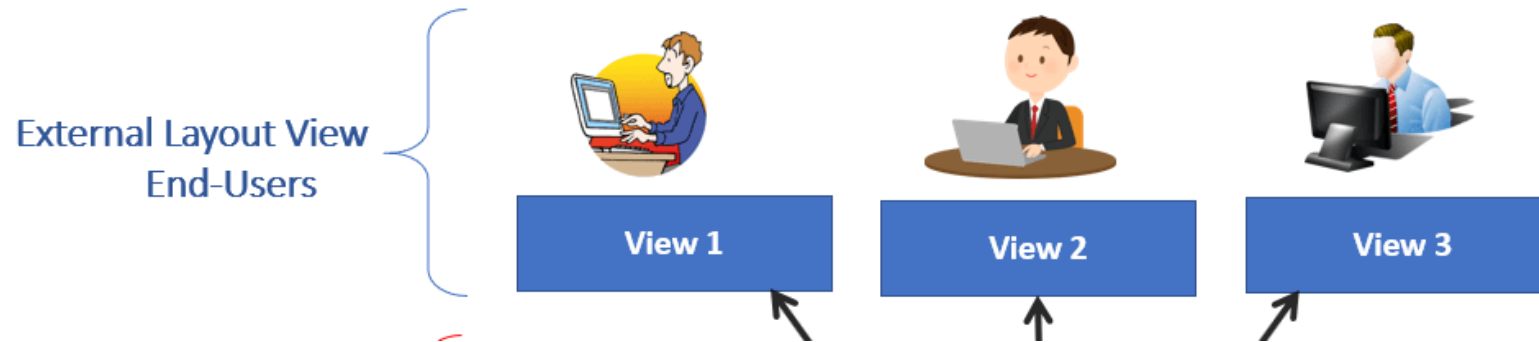
- Admin/IT:

What kind of information must be available for customers and cashiers- is this same?

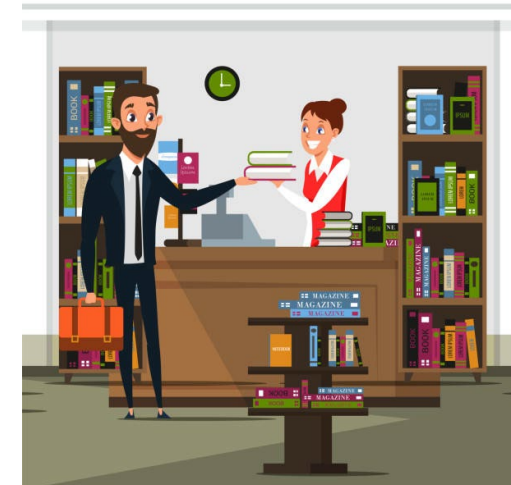
what information must be available for each user group.



DBMS-Three Level Architecture



- **Customers:** Look for books, place orders, view prices.
- **Cashiers:** Process sales, manage inventory, generate receipts.
- **Managers:** View reports, update stock levels, monitor sales trends.
- **Admin/IT:** Perform system maintenance, backup data.





Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

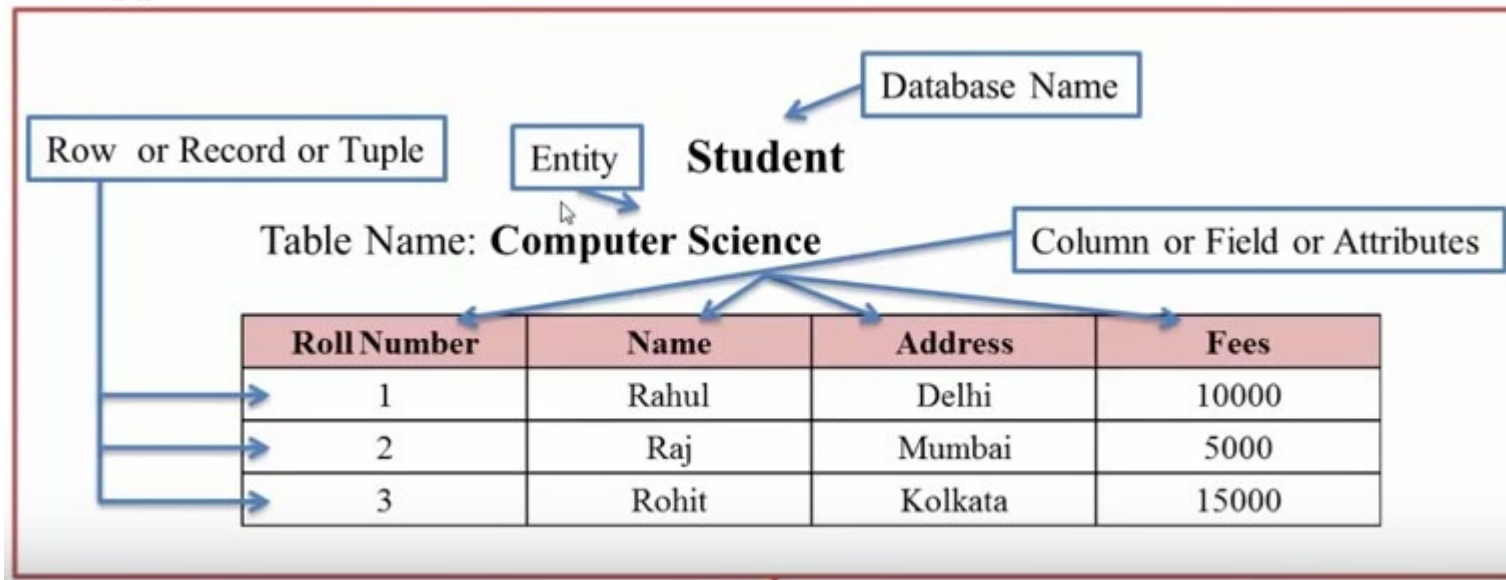
200
1822 - 2022

Relational Keys

Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

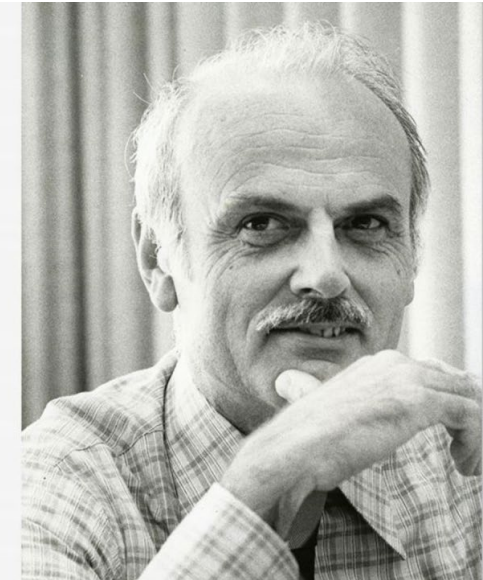
www.uwtsd.ac.uk

E/R (Entity/Relationship) Model



**Remembering
Edgar F. Codd**

The Father of Relational
Database Management
Systems



Key

- An attribute or group of attributes, which is used to identify a row/record in a relation.

- Different Types of Database Keys
 - Candidate Key
 - Primary Key
 - Super Key
 - Alternate Key
 - Foreign Key
 - Composite Key

Key

- An attribute or group of attributes, which is used to identify a row in a relation.

Different Types of Database Keys

- Candidate Key
- Primary Key
- ~~Super Key~~
- ~~Alternate Key~~
- Foreign Key
- Composite Key

Candidate Key

- set of one or more attributes (columns) in a relational database table that can uniquely identify a record (row) in that table.
- Each candidate key can serve as a unique identifier for the data, and at least one of these candidate keys will be chosen as the primary key for the table.

Consider a **Student** table with the following attributes:

StudentID	StudentNumber	Email	StudentName
1	2837590	adele@example.com	Adele
2	2837591	bob@example.com	Bob
3	2837592	charlie@example.com	Charlie

Candidate Key

- set of one or more attributes (columns) in a relational database table that can uniquely identify a record (row) in that table.
- Each candidate key can serve as a unique identifier for the data, and at least one of these candidate keys will be chosen as the primary key for the table.

Consider a **Student** table with the following attributes:

StudentID	StudentNumber	Email	StudentName
1	2837590	adele@example.com	Adele
2	2837591	bob@example.com	Bob
3	2837592	charlie@example.com	Charlie

Primary Keys

Consider a **Student** table with the following attributes:

- specific choice of a candidate key in a relational database that uniquely identifies each record (row) in a table

StudentID	StudentNumber	Email	StudentName
1	2837590	adele@example.com	Adele
2	2837591	bob@example.com	Bob
3	2837592	charlie@example.com	Charlie

Rules for defining Primary key:

- Two rows can't have the same primary key value
- It's must for every row to have a primary key value.
- The primary key field cannot be null.
- Once assigned, the values of a primary key should not change, as this could affect the integrity of the database.

Table: Student

Identify Candidate Keys

StudentName	StudentNumber	Email	DateOfBirth
Alice	2837590	alice@example.com	01-Jan-1995
Bob	2837591	bob@example.com	02-Feb-1996
Charlie	2837592	charlie@example.com	03-Mar-1997
Alice	2837593	alice@uni.edu	01-Jan-1995

Choose a Primary Key

Table: Course

CourseID	CourseName	Instructor	CourseCode
101	Database Design	Dr. Smith	DB101
102	Web Development	Prof. Johnson	WD102
103	Data Science	Dr. Lee	DS103

Foreign Key

- A foreign key is an attribute or a set of attributes in one table that refers to the primary key in another table.
- The primary purpose of a foreign key is to maintain referential integrity between the two tables.
- This means that a foreign key in one table must match an existing primary key in another table, ensuring that relationships between records remain consistent.

Tables:

1.Customer Table

CustomerID	CustomerName	Country
1	Alice	USA
2	Bob	UK

2.Order Table

OrderID	CustomerID	OrderDate
101	1	2024-01-15
102	2	2024-01-16
103	1	2024-01-17

Key Characteristics of Foreign Keys

- **Links Tables:** A foreign key connects two tables, showing how they relate to each other.
- **Keeps Data Accurate:** It makes sure that the data in one table matches the data in another table, preventing mistakes.
- **Same Data Type:** The foreign key must have the same type of data as the primary key it points to.
- **Can Have Many:** A table can have multiple foreign keys linking it to different tables.
- **Optional or Required:** A foreign key can be optional (can be empty) or required (must have a value).
- **Represents Relationships:** They show how different pieces of data are related, helping to keep everything organized.

Foreign key

Which is the foreign key here?

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Foreign key

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Primary key

Foreign key

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Composite Key

- Combination of two or more attributes (columns) in a database table that together uniquely identify a record

Consider a **Course Enrollment** table that records which students are enrolled in which courses.

•**Table Structure:**

StudentID	CourseID	EnrollmentDate
1	101	2024-01-10
1	102	2024-01-15
2	101	2024-01-12
3	103	2024-01-20

Composite Key

- Combination of two or more attributes (columns) in a database table that together uniquely identify a record

Consider a **Course Enrollment** table that records which students are enrolled in which courses.

•**Table Structure:**

StudentID	CourseID	EnrollmentDate
1	101	2024-01-10
1	102	2024-01-15
2	101	2024-01-12
3	103	2024-01-20

Keys

order

Identify the following keys from this order table

- Candidate Key?
- Primary Key?

OrderID	CustomerID	OrderDate	Invoice Number	ShippingAddress
301	501	2023-10-15	12345	123 Main St, City1
302	502	2023-10-16	54321	456 Elm St, City2
303	503	2023-10-17	98765	789 Oak St, City3

Keys

Identify the following keys from this Order table

- Candidate Key?

orderID,

Invoice number,

(customerID, orderDate)

- Primary Key?

orderID,

order

OrderID	CustomerID	OrderDate	Invoice Number	ShippingAddress
301	501	2023-10-15	12345	123 Main St, City1
302	502	2023-10-16	54321	456 Elm St, City2
303	503	2023-10-17	98765	789 Oak St, City3

Identify the composite key from this viewing table

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-13	too small
CR76	PG4	20-Apr-13	too remote
CR56	PG4	26-May-13	
CR62	PA14	14-May-13	no dining room
CR56	PG36	28-Apr-13	

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-13	too small
CR76	PG4	20-Apr-13	too remote
CR56	PG4	26-May-13	
CR62	PA14	14-May-13	no dining room
CR56	PG36	28-Apr-13	

ClientNo and propertyNo together form the composite key

Further Reading :

Chapter 2.1: Database Environment

Chapter 4.2.5: Relational Keys

- Database Systems: A Practical Approach to Design, Implementation, and Management Sixth edition by • Thomas Connolly • Carolyn Begg