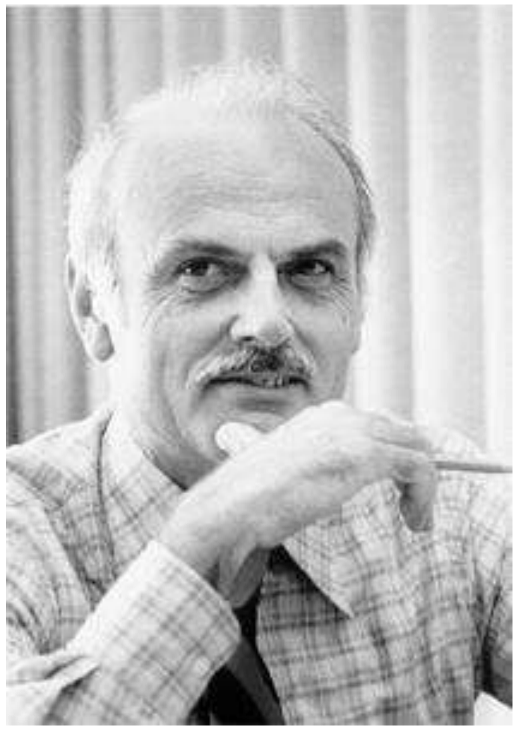
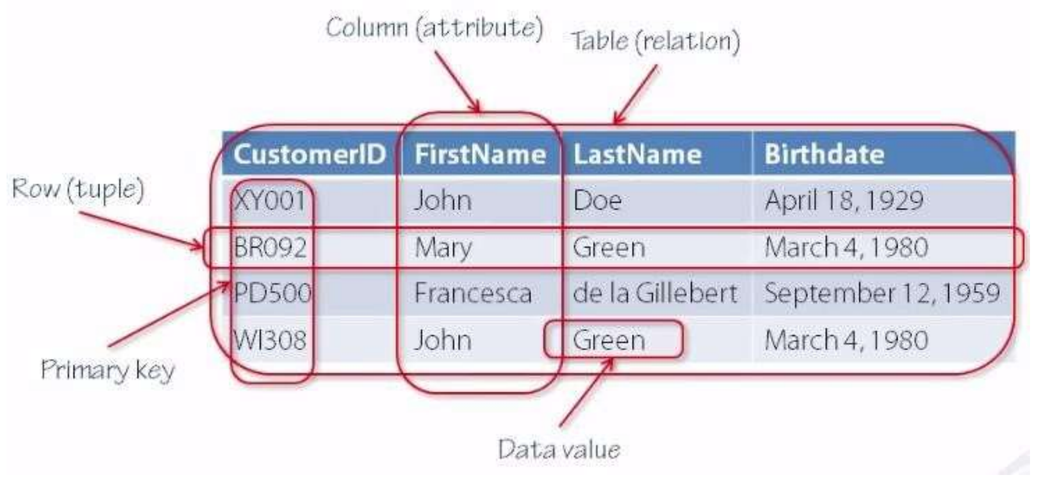
Functional Dependency

RELATIONAL DATABASE MANAGEMENT SYSTEM

- A relational database management system (RDBMS), conceptualized by Edgar F. Codd in 1970, serves as the foundation for most contemporary commercial and open-source database applications.
- Central to its design is the utilization of tables for data storage, where it maintains and enforces specific data relationships, marking a significant evolution in database design.





CustomerID	FirstName	LastName	Birthdate
XY001	John	Doe	April 18, 1929
BR092	Mary	Green	March 4, 1980
PD500	Francesca	de la Gillebert	September 12, 1959
WI308	John	Green	March 4, 1980

BASICS OF RDBMS

- **Domain (set of permissible value in particular column)** is a set of atomic values.
- By **atomic** we mean that each value in the domain is indivisible as far as the formal relational model is concerned.
- A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn.
  - **E.g.** Names: The set of character strings that represent names of persons.

Domain/ Field/ Column/ Arity/ Degree

NAME	ID	CITY	COUNTRY	HOBBY
NISHA	1	AGRA	INDIA	PLAYING
NIKITA	2	DELHI	INDIA	DANCING
AJAY	3	AGRA	INDIA	CHESS
ARPIT	4	PATNA	INDIA	READING

- **Table (Relation)** - A Relation is a set of tuples/rows/entities/records.
- **Tuple** - Each row of a Relation/Table is called Tuple.
- **Arity/Degree** - No. of columns/attributes of a Relation. E.g. - Arity is 5 in Table Student.

No. of rows/tuples/record of a Relational instance. e.g.-Cardinality is 4 in table Student.

- **Cardinality** - No of rows/tuples/record of a Relational instance. E.g. - Cardinality is 4 in table Student.

Rows/Tuples/Record/ Cardinality

NAME	ID	CITY	COUNTRY	HOBBY
NISHA	1	AGRA	INDIA	PLAYING
NIKITA	2	DELHI	INDIA	DANCING
AJAY	3	AGRA	INDIA	CHESS
ARPIT	4	PATNA	INDIA	READING

Properties of Relational tables

- Cells contains atomic values
- Values in a column are of the same kind
- Each row is unique
- No two tables can have the same name in a relational schema.
- Each column has a unique name
- The sequence of rows is insignificant
- The sequence of columns is insignificant.

Problems in relational database

- **Update Anomalies-** Anomalies that cause redundant work to be done during insertion into and Modification of a relation and that may cause accidental loss of information during a deletion from a relation
  - **Insertion Anomalies**
  - **Modification Anomalies**
  - **Deletion Anomalies**

Roll no	name	Age	Br_code	Br_name	Br_hod_name
1	A	19	101	Cs	Abc
2	B	18	101	Cs	Abc
3	C	20	101	Cs	Abc
4	D	20	102	Ec	Pqr

Insertion anomalies: An independent piece of information cannot be recorded into a relation unless an irrelevant information must be inserted together at the same time.

suppose a new branch was opened with name 'Mat' under Br_hod_name = 'Klm' with Br_code = '103'
then you would have to wait until a student is enrolled in this new branch or you would just fill the other rows with null.

Modification anomalies: The update of a piece of information must occur at multiple locations.

suppose the Br_name 'Cs' is now 'Cse', then you'd need to modify 3 rows.

Roll no	name	Age	Br_code	Br_name	Br_hod_name
1	A	19	101	Cs	Abc
2	B	18	101	Cs	Abc
3	C	20	101	Cs	Abc
4	D	20	102	Ec	Pqr

Deletion Anomalies: The deletion of a piece of information unintentionally removes other information.

suppose Roll_no = '4' was unenrolled, then deleting this record will also delete the info. of branch 'Ec'

Roll no	name	Age	Br_code	Br_name	Br_hod_name
1	A	19	101	Cs	Abc
2	B	18	101	Cs	Abc
3	C	20	101	Cs	Abc
4	D	20	102	Ec	Pqr

Roll no	name	Age	Br_code	Br_name	Br_hod_name
1	A	19	101	Cs	Abc
2	B	18	101	Cs	Abc
3	C	20	101	Cs	Abc
4	D	20	102	Ec	Pqr

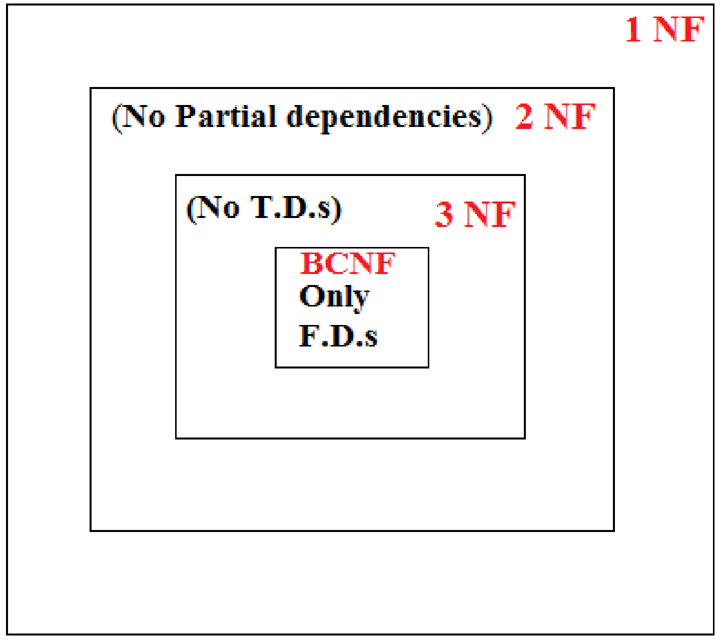
PK FK PK

Roll no	name	Age	Br_code
1	A	19	101
2	B	18	101
3	C	20	101
4	D	20	102

Br_code	Br_name	Br_hod_name
101	Cs	Abc
102	ec	Pqr

Purpose of Normalization

- Normalization may be simply defined as refinement process. Which includes creating tables and establishing relationships between those tables according to rules designed both to protect data and make the database more flexible by eliminating two factors;
    - Redundancy
    - Inconsistent Dependency
- With out normalization data base system may be inaccurate, slow and inefficient and they might not produce the data we expect. A series of normal form tests that can be carried out on individual relation schemas so that the relational database can be **normalized** to any desired degree.
- 1NF>>>2NF>>3NF>>BCNF



Hierarchy of Normal forms

FUNCTIONAL DEPENDENCY

- A formal tool for analysis of relational schemas.
- In a Relation R, if ‘α’ ⊑ R AND ‘β’ ⊑ R,
  then
  attribute or a Set of attribute ‘α’ Functionally derives an attribute or set of attributes ‘β’, iff each ‘α’ value is associated with precisely one ‘β’ value.
- For all pairs of tuples t_1 and t_2 in R such that
  - If $T_1[\alpha] = T_2[\alpha]$
  - Then, $T_1[\beta] = T_2[\beta]$

X	Y	Z
1	4	2
1	4	3
2	6	3
3	2	2

X->Y, X-|->Z, Y->X, Y-|->Z, Z-|->X,
Z-|->Y, XY-|->Z, YZ->X, XZ->Y

Q Consider the following relation instance, which of the following dependency doesn't hold

A) A → b

X B) BC → A

C) B → C

D) AC → B

A	B	C
1	2	3
4	2	3
5	3	3

Trivial Functional dependency - If β is a subset of α, then the functional dependency α → β will always hold.

X	Y	Z
1	4	2
1	4	3
2	6	3
3	2	2

- **ATTRIBUTES CLOSURE/CLOSURE ON ATTRIBUTE SET/ CLOSURE SET OF ATTRIBUTES**
- Attribute closure of an attribute set can be defined as set of attributes which can be functionally determined from F. DENOTED BY F^+

R (A, B, C, D)
A → B ✓
B → C ✓
AB → D ✓
A^+ = ABCD
CK
B^+ = BC

R(ABCDEFG)
A → B
BC → DE
AEG → G //
$(AC)^+$ =?
ACBDE

Q R(ABCDE)
A → BC
CD → E
B → D
E → A
$(B)^+$ = BD

ARMSTRONG’S AXIOMS

- An axiom or postulate is a statement that is taken to be true, to serve as a premise or starting point for further reasoning and arguments.
- Armstrong's axioms are a set of axioms (or, more precisely, inference rules) used to infer all the functional
- dependencies on a relational database. They were developed by William W. Armstrong in his 1974 paper.
- The axioms are sound in generating only functional dependencies in the closure of a set of functional dependencies (denoted as F+) when applied to that set (denoted as F).

Armstrong Axioms

- **Reflexivity**: If *Y* is a subset of *X*, then *X* → *Y*
- **Augmentation**: If *X* → *Y*, then *XZ* → *YZ*
- **Transitivity**: If *X* → *Y* and *Y* → *Z*, then *X* → *Z*

$\cancel{\ }$

ABC

subset: A, B, C, . . .

ABC → A

ABC → BC

XA → BX

From these rules, we can derive these secondary rules-

- **Union**: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
- **Decomposition**: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
- **Pseudo transitivity**: If $X \rightarrow Y$ and $WY \rightarrow Z$, then $WX \rightarrow Z$

$WX \rightarrow Y \quad W \rightarrow Z \Rightarrow WX \rightarrow Z$

- **Composition**: If $X \rightarrow Y$ and $Z \rightarrow W$, then $XZ \rightarrow YW$

Why Armstrong axioms refers to the Sound and Complete

- By sound, we mean that given a set of functional dependencies F specified on a relation schema R, any dependency that we can infer from F by using the primary rules of Armstrong axioms holds in every relation state r of R that satisfies the dependencies in F.
- By complete, we mean that using primary rules of Armstrong axioms repeatedly to infer dependencies until no more dependencies can be inferred results in the complete set of all possible dependencies that can be inferred from F.

Equivalence of Two FD sets-

Two FD sets F_1 and F_2 are equivalent if –

$F_1^+ = F_2^+$

Or

$F_1 \sqsubseteq F_2^+$ and $F_2 \sqsubseteq F_1^+$

Q Consider the following set of fd R(ACDEH)

F	G
A → C AC → D E → AD E → H	A → CD E → AH

STEP 1: RHS must be single [do decomposition]
STEP 2: if an FD can be gotten from other FDs, then the former FD is redundant. therefore, eliminate it.
STEP 3: LHS should be single if possible. if an attribute on the LHS can be gotten from other attributes in the LHS, then that attribute is eliminated.

To find the MINIMAL COVER / CANONICAL COVER / IRREDUCIBLE SET

- A **canonical cover** (also known as a minimal cover) for a set of functional dependencies in a database is a minimal set of functional dependencies that is equivalent to the original set, but with redundant dependencies and extraneous attributes removed. It is used in the normalization process of database design to simplify the set of functional dependencies and to find a good set of relations.

- There may be any following type of redundancy in the set of functional dependencies: -
  - Complete production may be Redundant.
  - One or more than one attributes may be redundant on right hand side of a production.
  - One or more than one attributes may be redundant on Left hand side of a production.

$C^+ = CB$ $A^+ = AB$

Q R(ABCD)

A → B

C → B

D → ABC

AC → D

canonical cover:
A->B
C->B
D->A –
D->C –
AC->D /

$D^+ = AB$

$AC^+ = ACB$

$D \to A$

$D \to B$

$D \to C$

$D^+ = DACB$

$\overline{R(A,B,C)}$

A → B

B → C

A → C

AB → B

AB → C

AC → B

canonical cover:

B->C

A->B

C->B

SK

CK

PK

CK

SK $\left\{ \boxed{A^+} = \overline{ABC} \right.$

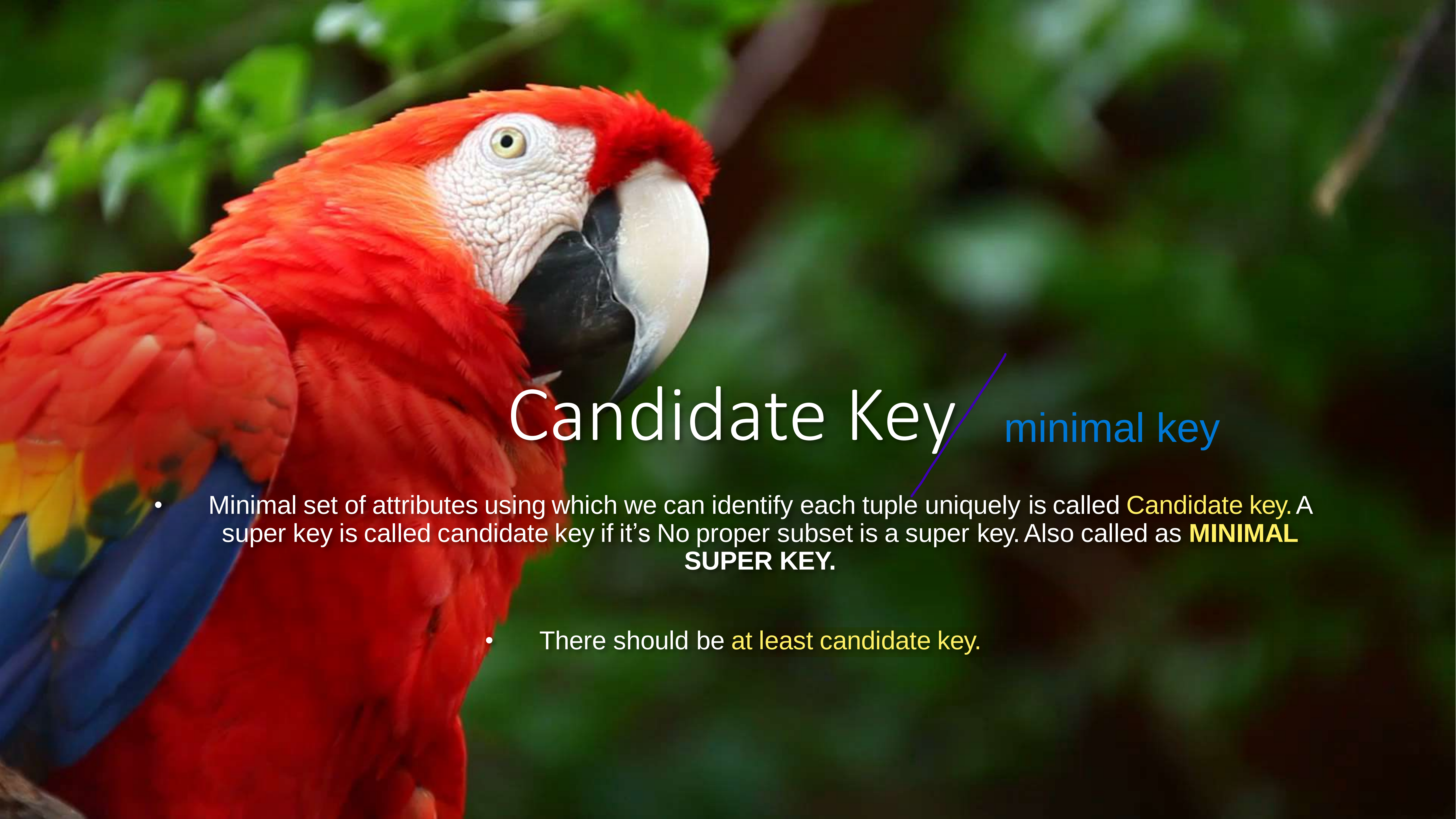
AB, AC, ~~ABC~~

Pr. A

AB/BC/CA

Super key

- Set of attributes using which we can identify each tuple uniquely is called Super key.
- Let X be a set of attributes in a Relation R , if X^+(Closure of X) determines all attributes of R then X is said to be Super key of R .
  X can be single or multiple attribute
- There should be at least one Super key in every relation.



Candidate Key

minimal key

- Minimal set of attributes using which we can identify each tuple uniquely is called Candidate key. A super key is called candidate key if it's No proper subset is a super key. Also called as **MINIMAL SUPER KEY.**

- There should be at least candidate key.

Prime attribute - Attributes that are member of at least one candidate Keys are called Prime attributes.

Primary key

- One of the candidate keys is selected by database administrator as a Primary means to identify tuple is called primary Key. Primary Key attribute are not allowed to have Null values. Exactly one Primary Key per table in RDMS.

- Candidate key which are not chosen as primary key is alternate key.

Foreign Keys

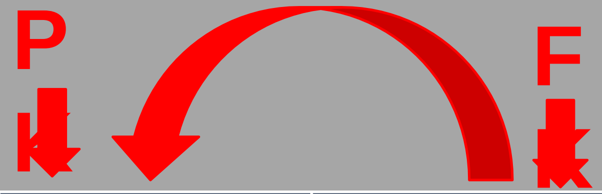
- A foreign key is a column or group of columns in a relational database table that refers the primary key of the same table or some other table to represent relationship.
- The ***concept of referential integrity*** is derived from foreign key theory.

Roll no	name	Age	Br_code	Br_name	Br_hod_name
1	A	19	101	Cs	Abc
2	B	18	101	Cs	Abc
3	C	20	101	Cs	Abc
4	D	20	102	Ec	Pqr

PK FK PK

Roll no	name	Age	Br_code
1	A	19	101
2	B	18	101
3	C	20	101
4	D	20	102

Br_code	Br_name	Br_hod_name
101	Cs	Abc
102	ec	Pqr



Roll no	CR
1	1
2	2
3	1
4	2
5	1
6	2
7	1
8	2
9	1
10	2
11	1
12	2
13	1
14	2
15	null

Composite key – Composite key is a key composed of more than one column sometimes it is also known as concatenated key.

Secondary key – Secondary key is a key used to speed up the search and retrieval contrary to primary key, a secondary key does not necessary contain unique values.

example: name