UNIT-2

Overview of RDBMS

Outlines…

- Relational Database Concepts
- Integrity
- Normalization

Relational Database Concept

- A database is a structured collection of data related to some real-life phenomena that we are trying to model. A relational database is one where the database structure is in the form of tables. Formally, a relation R defined over n sets D1;D2; : : : ;Dn (not necessarily distinct) is a set of n-tuples (or simply tuples) hd1;d2; : : : ;dni such that d1 2 D1;d2 2 D2; : : : ;dn 2 Dn.

Example

- As an example we use a database that models an engineering company. The entities to be modeled are the employees (EMP) and projects (PROJ). For each employee, we would like to keep track of the employee number (ENO), name (ENAME), title in the company (TITLE), salary (SAL), identification number of the project(s) the employee is working on (PNO), responsibility within the project (RESP), and duration of the assignment to the project (DUR) in months. Similarly, for each project we would like to store the project number (PNO), the project name (PNAME), and the project budget (BUDGET).
- EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR

- PROJ

PNO	PNAME	BUDGET

The relation schemas

- EMP(ENO, ENAME, TITLE, SAL, PNO, RESP, DUR)
- PROJ(PNO,PNAME, BUDGET)

- In relation scheme EMP, there are seven attributes: ENO, ENAME, TITLE, SAL, PNO, RESP, DUR. The values of ENO come from the domain of all valid employee numbers, say D1, the values of ENAME come from the domain of all valid names, say D2, and so on. Note that each attribute of each relation does not have to come from a distinct domain. Various attributes within a relation or from a number of relations may be defined over the same domain.

Key Relation schema

- The key of a relation scheme is the minimum non-empty subset of its attributes such that the values of the attributes comprising the key uniquely identify each tuple of the relation.
- The attributes that make up key are called prime attributes.
- The superset of a key is usually called a super key.
- Thus in our example the key of PROJ is PNO, and that of EMP is the set (ENO, PNO). Each relation has at least one key.
- Sometimes, there may be more than one possibility for the key. In such cases, each alternative is considered a candidate key, and one of the candidate keys is chosen as the primary key, which we denote by underlining.
- The number of attributes of a relation defines its degree, whereas the number of tuples of the relation defines its cardinality.

Key Relation schema

- In tabular form, the example database consists of two tables, as shown in Figure. The columns of the tables correspond to the attributes of the relations; **if there were any information entered as the rows, they would correspond to the tuples**.
- The empty table, showing the structure of the table, corresponds to the **relation schema**.
- **when the table is filled with rows, it corresponds to a relation instance**.
- Since the information within a table varies over time, many instances can be generated from one relation scheme.
- Note that from now on, the term relation refers to a relation instance. In Figure we depict instances of the **two relations** that are defined.

Example Relation Instances

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

Repetition Anomaly

- The NAME,TITLE, SAL attribute values are repeated for each project that the employee is involved in.
  - Waste of space
  - Complicates updates

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

Update Anomaly

- If any attribute of project (say SAL of an employee) is updated, multiple tuples have to be updated to reflect the change.

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

Insertion Anomaly

- It may not be possible to store information about a new project until an employee is assigned to it.

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

Deletion Anomaly

- If an engineer, who is the only employee on a project, leaves the company, his personal information cannot be deleted, or the information about that project is lost.
- May have to delete many tuples.

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

What to do?

- Take each relation individually and "improve" it in terms of the desired characteristics
  - Normal forms
    - Atomic values (1NF)
    - Can be defined according to keys and dependencies.
    - Functional Dependencies (2NF, 3NF, BCNF)
    - Multivalued dependencies (4NF)
  - Normalization
    - Normalization is a process of concept separation which applies a top-down methodology for producing a schema by *subsequent refinements and decompositions*.
    - Do not combine unrelated sets of facts in one table; each relation should contain an independent set of facts.
    - Universal relation assumption
    - 1NF to 3NF; 1NF to BCNF

Normalization Issues

- How do we decompose a schema into a desirable normal form?
- What criteria should the decomposed schemas follow in order to preserve the semantics of the original schema?
  - Reconstructability: recover the original relation $\Rightarrow$ no spurious joins
  - Lossless decomposition: no information loss
  - Dependency preservation: the constraints (i.e., dependencies) that hold on the original relation should be enforceable by means of the constraints (i.e., dependencies) defined on the decomposed relations.
- What happens to queries?
  - Processing time may increase due to joins
  - Denormalization

Functional Dependence

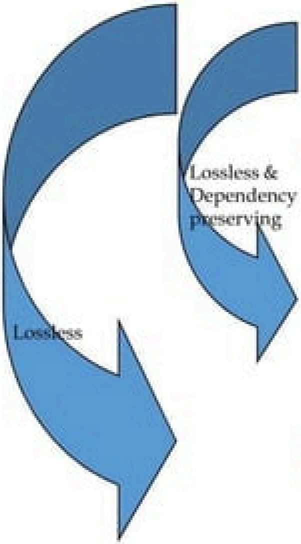
- Given relation R defined over $U = \{A_1, A_2, ..., A_n\}$ where $X \subseteq U$, $Y \subseteq U$. If, for all pairs of tuples t_1 and t_2 in any legal instance of relation scheme R,

  $$t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y],$$

  then the functional dependency $X \rightarrow Y$ holds in R.
- Example
  - In relation EMP
    - (ENO, PNO) $\rightarrow$ (ENAME, TITLE, SAL, DUR, RESP)
  - In relation PROJ
    - PNO $\rightarrow$ (PNAME, BUDGET)

Normal Forms Based on FDs

1NF eliminates the relations within relations or relations as attributes of tuples.



First Normal Form (1NF)

↓ eliminate the partial functional dependencies of non-prime attributes to key attributes

Second Normal Form (2NF)

↓ eliminate the transitive functional dependencies of non-prime attributes to key attributes

Third Normal Form (3NF)

↓ eliminate the partial and transitive functional dependencies of prime (key) attributes to key.

Boyce-Codd Normal Form (BCNF)

Normalized Relations – Example

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Relational Algebra

Specify how to obtain the result using a set of operators

Form

$$\langle Operator \rangle_{\langle parameters \rangle} \langle Operands \rangle \rightarrow \langle Result \rangle$$

$$\underset{\text{Relation (s)}}{\downarrow} \qquad \underset{\text{Relation}}{\downarrow}$$

Relational Algebra Operators

- Fundamental
  - Selection
  - Projection
  - Union
  - Set difference
  - Cartesian product
- Additional
  - Intersection
  - θ-join
  - Natural join
  - Semijoin
  - Division
- Union compatibility
  - Same degree
  - Corresponding attributes defined over the same domain

Selection

- Produces a horizontal subset of the operand relation
- General form

$$\sigma_F(R)=\{t \mid t \in R \text{ and } F(t) \text{ is true}\}$$

where

- R is a relation, t is a tuple variable
- F is a formula consisting of
  - operands that are constants or attributes
  - arithmetic comparison operators

    $<, >, =, \neq, \leq, \geq$
  - logical operators

    $\wedge, \vee, \neg$

Selection Example

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

$\sigma_{TITLE='Elect. Eng.'}(EMP)$

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E6	L. Chu	Elect. Eng.

Projection

- Produces a vertical slice of a relation
- General form

$$\Pi_{A_1,\ldots,A_n}(R)=\{t[A_1,\ldots, A_n] \mid t\in R\}$$

  where
  - R is a relation, t is a tuple variable
  - $\{A_1,\ldots, A_n\}$ is a subset of the attributes of R over which the projection will be performed
- Note: projection can generate duplicate tuples. Commercial systems (and SQL) allow this and provide
  - Projection with duplicate elimination
  - Projection without duplicate elimination

Projection Example

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

$\Pi_{PNO,BUDGET}(PROJ)$

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000

Union

- Similar to set union
- General form

$$R \cup S = \{t \mid t \in R \text{ or } t \in S\}$$

where R, S are relations, t is a tuple variable

  - Result contains tuples that are in R or in S, but not both (duplicates removed)
  - R, S should be union-compatible

Set Difference

- General Form

$$R - S = \{t \mid t \in R \text{ and } t \notin S\}$$

where R and S are relations, t is a tuple variable

  - Result contains all tuples that are in R, but not in S.
  - $R - S \neq S - R$
  - R, S union-compatible

Cartesian (Cross) Product

- Given relations
  - R of degree k_1, cardinality n_1
  - S of degree k_2, cardinality n_2
- Cartesian (cross) product:

$$R \times S = \{t\,[A_1,\ldots,A_{k_1}, A_{k_1+1},\ldots,A_{k_1+k_2}] \mid t[A_1,\ldots,A_{k_1}] \in R \text{ and } t[A_{k_1+1},\ldots,A_{k_1+k_2}] \in S\}$$

The result of $R \cdot S$ is a relation of degree $(k_1 + k_2)$ and consists of all $(n_1 \cdot n_2)$-tuples where each tuple is a concatenation of one tuple of R with one tuple of S.

Cartesian Product Example

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

PAY

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	45000
Programmer	60000

EMP × PAY

ENO	ENAME	EMP.TITLE	PAY.TITLE	SALARY
E1	J. Doe	Elect. Eng.	Elect. Eng.	55000
E1	J. Doe	Elect. Eng.	Syst. Anal.	70000
E1	J. Doe	Elect. Eng.	Mech. Eng.	45000
E1	J. Doe	Elect. Eng.	Programmer	60000
E2	M. Smith	Syst. Anal.	Elect. Eng.	55000
E2	M. Smith	Syst. Anal.	Syst. Anal.	70000
E2	M. Smith	Syst. Anal.	Mech. Eng.	45000
E2	M. Smith	Syst. Anal.	Programmer	60000
E3	A. Lee	Mech. Eng.	Elect. Eng.	55000
E3	A. Lee	Mech. Eng.	Syst. Anal.	70000
E3	A. Lee	Mech. Eng.	Mech. Eng.	45000
E3	A. Lee	Mech. Eng.	Programmer	60000
E8	J. Jones	Syst. Anal.	Elect. Eng.	55000
E8	J. Jones	Syst. Anal.	Syst. Anal.	70000
E8	J. Jones	Syst. Anal.	Mech. Eng.	45000
E8	J. Jones	Syst. Anal.	Programmer	60000

Intersection

- Typical set intersection

$$R \cap S \quad = \{t \mid t \in R \text{ and } t \in S\}$$

$$= R - (R - S)$$

- R, S union-compatible

θ–Join

- General form

$R \bowtie_{F(R.A_i, S.B_j)} S = \{t[A_1,\ldots,A_n, B_1,\ldots,B_m] \mid$
$t[A_1,\ldots,A_n] \in R$ and $t[B_1,\ldots,B_m] \in S$
and $F(R.A_i, S.B_j)$ is true$\}$

where

  - R, S are relations, t is a tuple variable
  - $F(R.A_i, S.B_j)$is a formula defined as that of selection.
- A derivative of Cartesian product
  - $R \bowtie_F S = \sigma_F(R \times S)$

Join Example

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.
E9	A. Hsu	Programmer
E10	T. Wong	Syst. Anal.

(a)

EMP $\bowtie_{\text{EMP.ENO=ASG.ENO}}$ ASG

ENO	ENAME	TITLE	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	P1	Manager	12
E2	M. Smith	Syst. Anal.	P1	Analyst	12
E2	M. Smith	Syst. Anal.	P2	Analyst	12
E3	A. Lee	Mech. Eng.	P3	Consultant	12
E3	A. Lee	Mech. Eng.	P4	Engineer	12
E4	J. Miller	Programmer	P2	Programmer	12
E5	J. Miller	Syst. Anal.	P2	Manager	12
E6	L. Chu	Elect. Eng.	P4	Manager	12
E7	R. Davis	Mech. Eng.	P3	Engineer	12
E8	J. Jones	Syst. Anal.	P3	Manager	12

(b)

Types of Join

- Equi-join
  - The formula *F* only contains equality
  - $R \bowtie_{R.A=S.B} S$
- Natural join
  - Equi-join of two relations *R* and *S* over an attribute (or attributes) common to both *R* and *S* and projecting out one copy of those attributes
  - $R \bowtie S = \Pi_{R \cup S} \sigma_F (R \times S)$

Natural Join Example

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

PAY

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	45000
Programmer	60000

EMP $\bowtie$ PAY

ENO	ENAME	TITLE	SALARY
E1	J. Doe	Elect. Eng.	55000
E2	M. Smith	Analyst	70000
E3	A. Lee	Mech. Eng.	45000
E4	J. Miller	Programmer	60000
E5	B. Casey	Syst. Anal.	70000
E6	L. Chu	Elect. Eng.	55000
E7	R. Davis	Mech. Eng.	45000
E8	J. Jones	Syst. Anal.	70000

Join is over the common attribute TITLE

Types of Join

- Outer-Join
  - Ensures that tuples from one or both relations that do not satisfy the join condition still appear in the final result with other relation' s attribute values set to NULL
  - Left outer join ⟕
  - Right outer join ⟖
  - Full outer join ⟗

Outer Join Example

- Left outer join

$EMP ⟕_{ENO} ASG$

ENO	ENAME	TITLE	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	P1	Manager	12
E2	M. Smith	Syst. Anal.	P1	Analyst	12
E2	M. Smith	Syst. Anal.	P2	Analyst	12
E3	A. Lee	Mech. Eng.	P3	Consultant	12
E3	A. Lee	Mech. Eng.	P4	Engineer	12
E4	J. Miller	Programmer	P2	Programmer	12
E5	J. Miller	Syst. Anal.	P2	Manager	12
E6	L. Chu	Elect. Eng.	P4	Manager	12
E7	R. Davis	Mech. Eng.	P3	Engineer	12
E8	J. Jones	Syst. Anal.	P3	Manager	12
E9	A. Hsu	Programmer	Null	Null	Null
E10	T. Wong	Syst. Anal.	Null	Null	Null

Semijoin

- Derivation

  $R \ltimes_F S = \Pi_A(R \bowtie_F S) = \Pi_A(R) \bowtie \Pi_{A \cap B}(S) = R \bowtie_F \Pi_{A \cap B}(S)$

  where

  - *R*, *S* are relations
  - *A* is a set of attributes

Semijoin Example

$EMP \ltimes_{EMP.TITLE=PAY.TITLE} PAY$

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Analyst
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

Division (Quotient)

- Given relations
  - R of degree k_1 ($R = \{A_1,\ldots,A_{k_1}\}$)
  - S of degree k_2 ($S = \{B_1,\ldots,B_{k_2}\}$)

  Let $A = \{A_1,\ldots,A_{k_1}\}$ [i.e., $R(A)$] and $B = \{B_1,\ldots,B_{k_2}\}$ [i.e., $S(B)$] and $B \subseteq A$.
  Then, $T = R \div S$ gives T of degree k_1-k_2 [i.e., $T(Y)$ where $Y = A$-B] such that for a tuple t to appear in T, the values in t must appear in R in combination with *every tuple* in S.
- Derivation

  $R \div S = \Pi_Y(R) - \Pi_Y((\Pi_Y(R) \times S) - R)$

Division Example

ASG'

ENO	PNO	PNAME	BUDGET
E1	P1	Instrumentation	150000
E2	P1	Instrumentation	150000
E2	P2	Database Develop.	135000
E3	P3	CAD/CAM	250000
E3	P4	Maintenance	310000
E4	P2	Database Develop.	135000
E5	P2	Database Develop.	135000
E6	P4	Maintenance	310000
E7	P3	CAD/CAM	250000
E8	P3	CAD/CAM	250000

PROJ'

PNO	PNAME	BUDGET
P3	CAD/CAM	250000
P4	Maintenance	310000

(ASG' ÷ PROJ')

ENO
E3

Relational Calculus

- Specify the properties that the result should hold
- Tuple relational calculus
- Domain relational calculus

Tuple Relational Calculus

- Query of the form $\{t \mid F\{t\}\}$ where
  - t is a tuple variable
  - F is a well-formed formula
- Atomic formula
  - Tuple-variable membership expressions
    - $R.t$ or $R(t)$: tuple t belongs to relation R
  - Conditions
    - $s[A]\ \theta\ t[B]$; s and t are tuple variables, A and B are components of s and t, respectively, $\theta \in \{<, >, =, \neq, \leq, \geq\}$; e.g., s[SAL] > t[SAL]
    - $s[A]\ \theta\ c$; s, A, and θ as defined above, c is a constant; e.g., s[ENAME] = 'Smith'
- SQL is an example of tuple relational calculus (at least in its simple form)

Domain Relational Calculus

- Query of the form $x_1, x_2, \ldots, x_n \mid F(x_1, x_2, \ldots, x_n)$ where
  - F is a well-formed formula in which $x_1, x_2, \ldots, x_n$ are the free variables
- QBE is an example

EMP	ENO	ENAME	TITLE
	<u>E2</u>	P.	

ASG	ENO	PNO	RESP	DUR
	<u>E2</u>	<u>P3</u>		

PROJ	PNO	PNAME	BUDGET
	<u>P3</u>	CAD/CAM	

INTEGRITY RULES

- Integrity rules are constraints that define consistent state of the database. They are usually expressed as assertions. Integrity constraints can be structural or behavioral.
- Structural constraints are inherent to the data model in the sense that they captured information on data relationships that cannot be modeled directly.
- Behavioral constraints permit the capturing of the semantics of the applications.
- The dependencies discussed in the preceding section are behavioral constraints.
- According to the two minimal structural constraints of the relational model are the **entity rule** and **referential integrity rules**.

INTEGRITY RULES

- By definition, any relation has a primary key. The entity rule dictates that each attribute of the key is no null. In Example attribute PNO of relation PROJ and attributes ENO,PNO of relation EMP cannot have null values. This constraint is necessary to enforce the fact that keys are unique.
- **Referential integrity is useful for capturing relationships between objects that the relational model cannot represent.**
- Referential integrity involves two relation is the key of another relation.
- In example there can bee referential integrity constraint between relation PROJ and EMP on attribute PNO.
- This rules prescribe that each employee belong to at least one existing project.