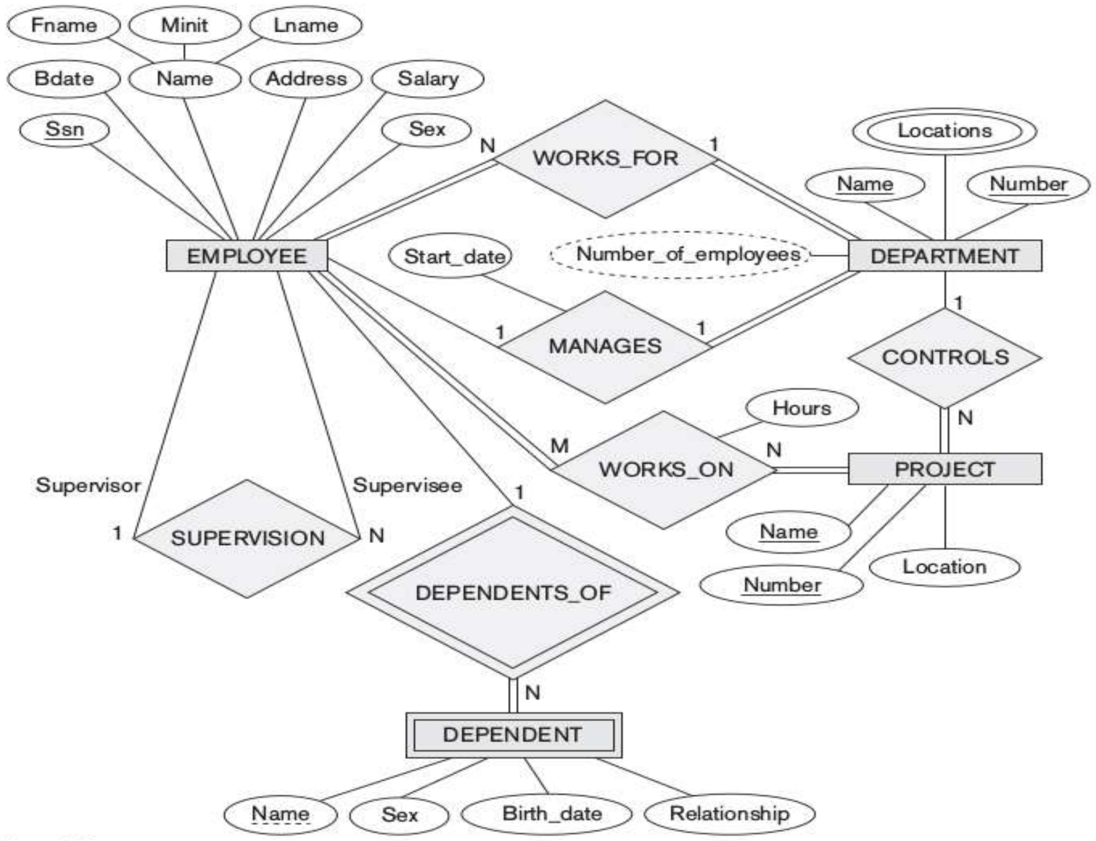
CS254 Database Management Systems Lec07

Instructor: Souradyuti Paul
souradyuti@iitbhilai.ac.in

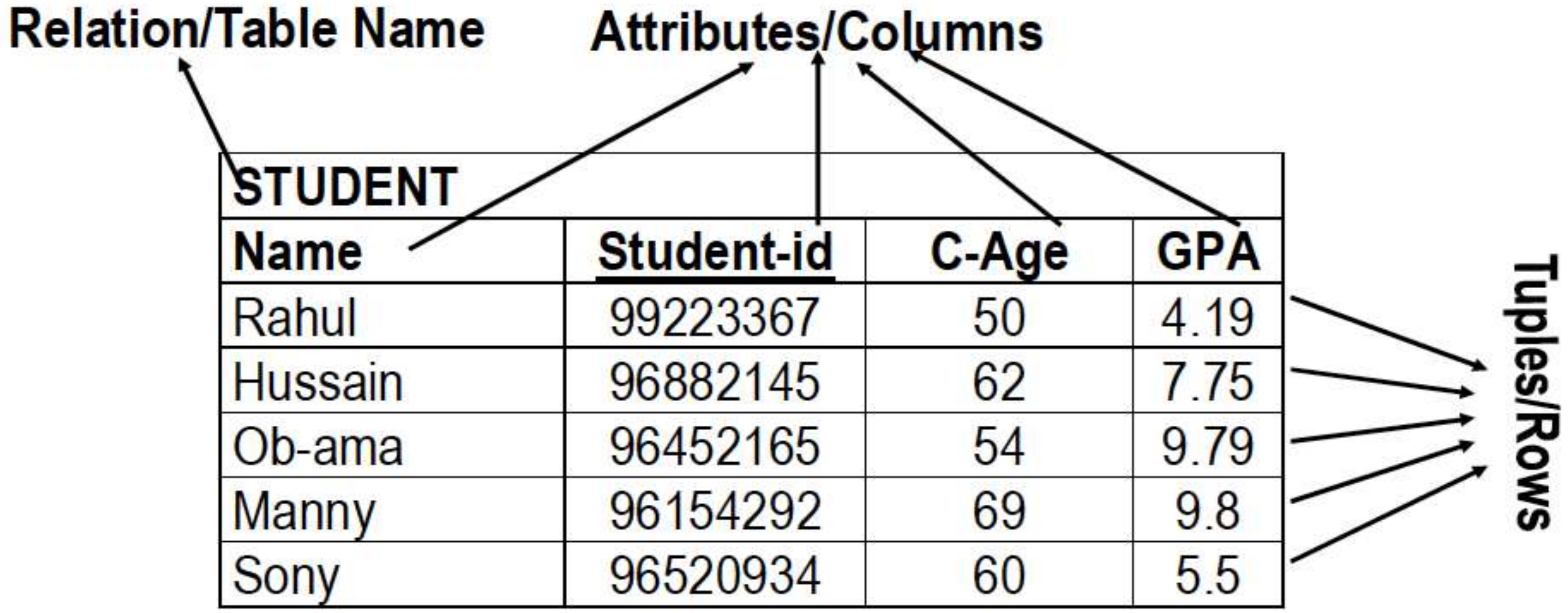
March 7, 2019


Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
Hours
N
M
N
WORKS_ON
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Flush ER and EER Models From Mind

Relational Data Model
&
Relational Algebra

Example: Student Relation



STUDENT			
Name	**Student-id**	**C-Age**	**GPA**
Rahul	99223367	50	4.19
Hussain	96882145	62	7.75
Ob-ama	96452165	54	9.79
Manny	96154292	69	9.8
Sony	96520934	60	5.5

Relational Model Concepts

Relation (informally): A table of values. Each column in the table has a column header called an attribute. Each row is called a tuple.

Formal Relational Concepts

–Domain: A set of atomic (indivisible) values.

–Attribute: A name to suggest the meaning that a domain plays in a particular relation. Each attribute A_i has a domain $dom(A_i)$.

–Relational Schema: A relation name R and a set of attributes A_i that define the relation.

Denoted by: $R(A_1, A_2, ..., A_n)$

Example: STUDENT(Name, Student-id, Age,GPA)

Relational Model Concepts(cont.)

Degree of a relation: Its number of attributes n.

Tuple t (of $R(A_1, A_2, ..., A_n)$): A (ordered) set of values $<v_1, v_2, ..., v_n>$ where each v_i is an element of $dom(A_i)$.

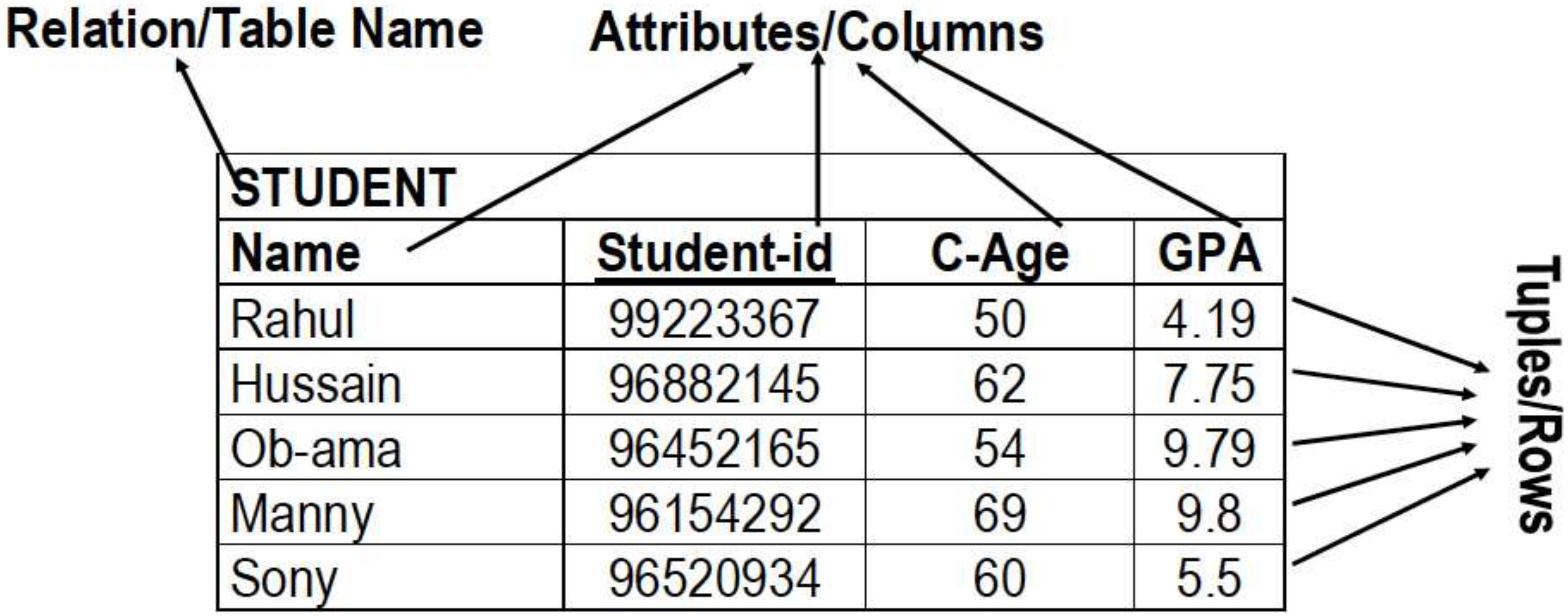
Also called an n-tuple.

Relation instance r(R): A set of tuples.

$r(R) = \{t_1, t_2, ..., t_m\}$, or alternatively

$r(R) \subseteq dom(A_1) \times dom(A_2) \times ... \times dom(A_n)$

Example: Student Relation



STUDENT			
Name	**Student-id**	**C-Age**	**GPA**
Rahul	99223367	50	4.19
Hussain	96882145	62	7.75
Ob-ama	96452165	54	9.79
Manny	96154292	69	9.8
Sony	96520934	60	5.5

Relational Model Concepts

Characteristics of Relations

- Ordering of tuples in a relation r(R): The tuples are not considered to be ordered, even though they appear to be in the tabular form
- Ordering of attributes in a relation schema R (and of values within each tuple): We will consider the attributes in $R(A_1, A_2, ..., A_n)$ and the values in $t=<v_1,v_2,... v_n>$ to be ordered.
- Values in a tuple: All values are considered atomic. A special null value is used to represent values that are unknown or inapplicable to certain tuples.

Integrity Constraints

Constraints are conditions that must hold on all valid relation instances. These constraints are Domain constraints, Key constraints, Entity integrity constraints, and Referential integrity constraints

Domain Constraints

Each attribute A must be an atomic value from the domain dom(A) for that attribute. The standard types of domain include integers, real numbers, characters, fixed length strings.

Key Constraints

Superkey of R: A set of attributes SK of R such that no two tuples in any valid relational instance r(R) will have same value for SK, i.e., for any distinct tuples t_1 and t_2 in r(R), $t_1[SK] \neq t_2[SK]$.

Key of R: A minimal superkey; that is a superkey K such that removal of any attribute from K results in a set of attributes that is not a superkey

Example: STUDENT relation has key {Student-id}, and superkey {Student-id,Name}.

If a relation has several candidate keys, one is chosen arbitrarily to be the primary key. Primary key attributes are underlined.

Referential Integrity

A constraint involving two relations (the previous constraints involve a single relation).

Used to specify a relationship among tuples in two relations: the referencing relation and the referenced relation.

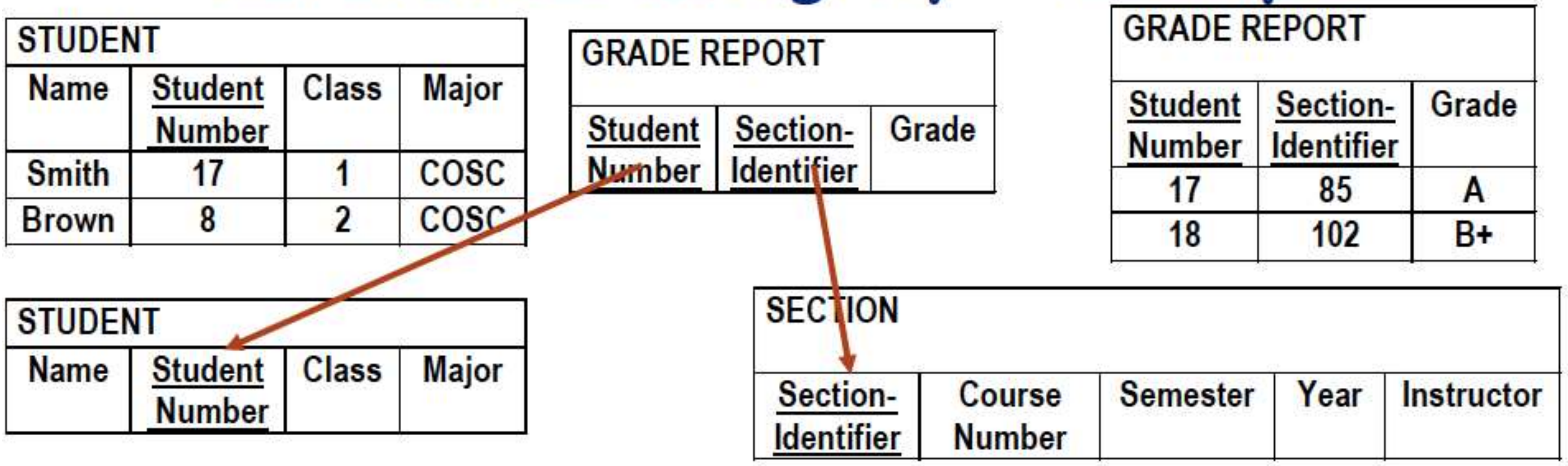
Tuples in the referencing relation R_1 have attributes FK (called foreign key attributes) that reference the PK (called primary key attributes) of the referenced relation R_2.

A tuple t_1 in R1 is said to reference a tuple t_2 in R_2

if t_1[FK] = t_2[PK]; Or it can be NULL, i.e., t_1[FK] = NULL

If the above condition is violated, then it implies that referential integrity constraint is violated.

Referential Integrity - Example



STUDENT

Name	Student Number	Class	Major
Smith	17	1	COSC
Brown	8	2	COSC

GRADE REPORT

Student Number	Section-Identifier	Grade

GRADE REPORT

Student Number	Section-Identifier	Grade
17	85	A
18	102	B+

STUDENT

Name	Student Number	Class	Major

SECTION

Section-Identifier	Course Number	Semester	Year	Instructor

SECTION

Section-Identifier	Course Number	Semester	Year	Instructor
85	MATH2410	Fall	91	King
92	COSC1310	Fall	91	Anderson
102	COSC3320	Spring	92	Knuth
135	COSC3380	Fall	92	Stone

A referential integrity constraint can be displayed in a relational database schema as a directed arc from R1.FK to R2.PK

Update Operations on Relations

Insert a tuple; Delete a tuple; Modify a tuple

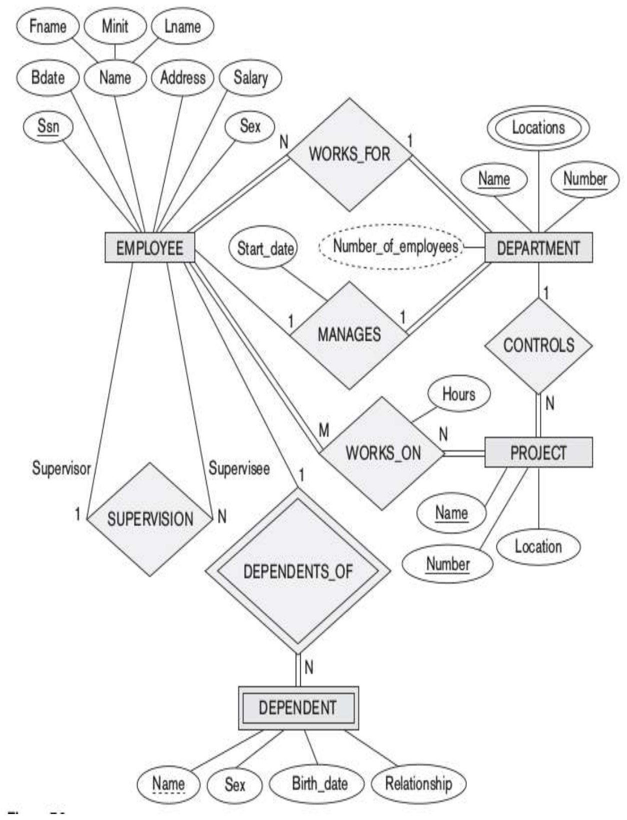
Integrity constraints should not be violated by the update operations

Several update operations may have to be grouped together.

Updates may propagate to cause other updates automatically. This may be necessary to maintain integrity constraints.

In case of integrity violation, several actions can be taken:

- cancel the operation
- perform the operation, but inform the user
- trigger additional updates so the violation is corrected
- execute a user-specified error-correction routine



COMPANY DATABASE

EMPLOYEE
FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO

DEPARTMENT
DNAME, DNUMBER, MGRSSN, MGRSTARTDATE

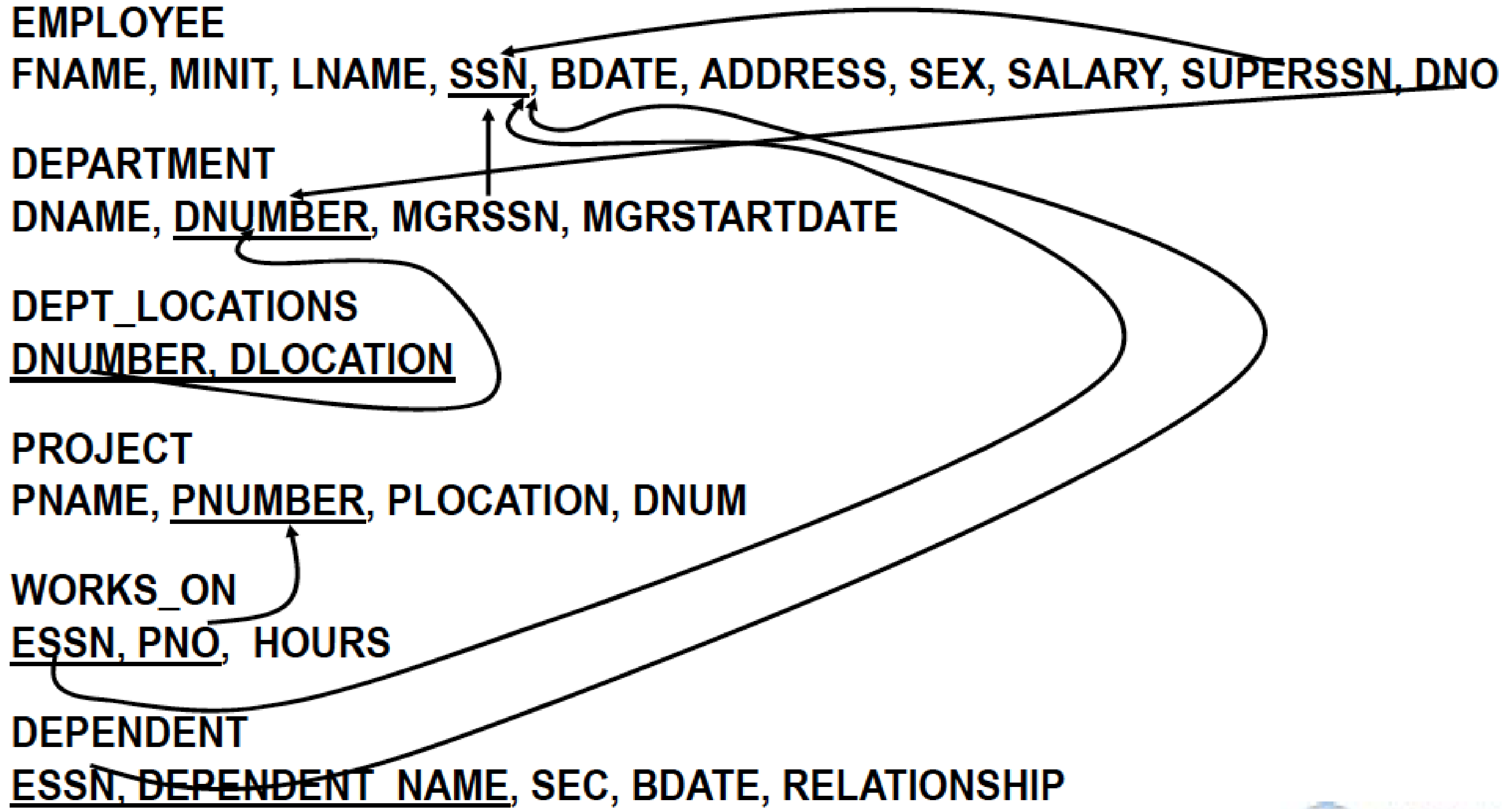
DEPT_LOCATIONS
DNUMBER, DLOCATION

PROJECT
PNAME, PNUMBER, PLOCATION, DNUM

WORKS_ON
ESSN, PNO, HOURS

DEPENDENT
ESSN, DEPENDENT_NAME, SEC, BDATE, RELATIONSHIP

COMPANY DATABASE



EMPLOYEE
FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO

DEPARTMENT
DNAME, DNUMBER, MGRSSN, MGRSTARTDATE

DEPT_LOCATIONS
DNUMBER, DLOCATION

PROJECT
PNAME, PNUMBER, PLOCATION, DNUM

WORKS_ON
ESSN, PNO, HOURS

DEPENDENT
ESSN, DEPENDENT_NAME, SEC, BDATE, RELATIONSHIP

ER-to-Relational Mapping

1 For each regular entity in the ER schema, create a relation R that includes all the simple attributes of E. Choose one of the key attributes of E as primary key for R.

2 For each binary relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.

- For 1:1 relationship b/w S and T:: choose one of the relations -- S say, and include as foreign key in S the primary key of T.
- For 1:N relationship b/w S and T:: let S be entity participating at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T.
- For M:N relationship b/w S and T:: create a new relation P to represent R, and include as foreign key attributes in P the primary keys of the relations (i.e. of S and T) that represent the participating entity types.

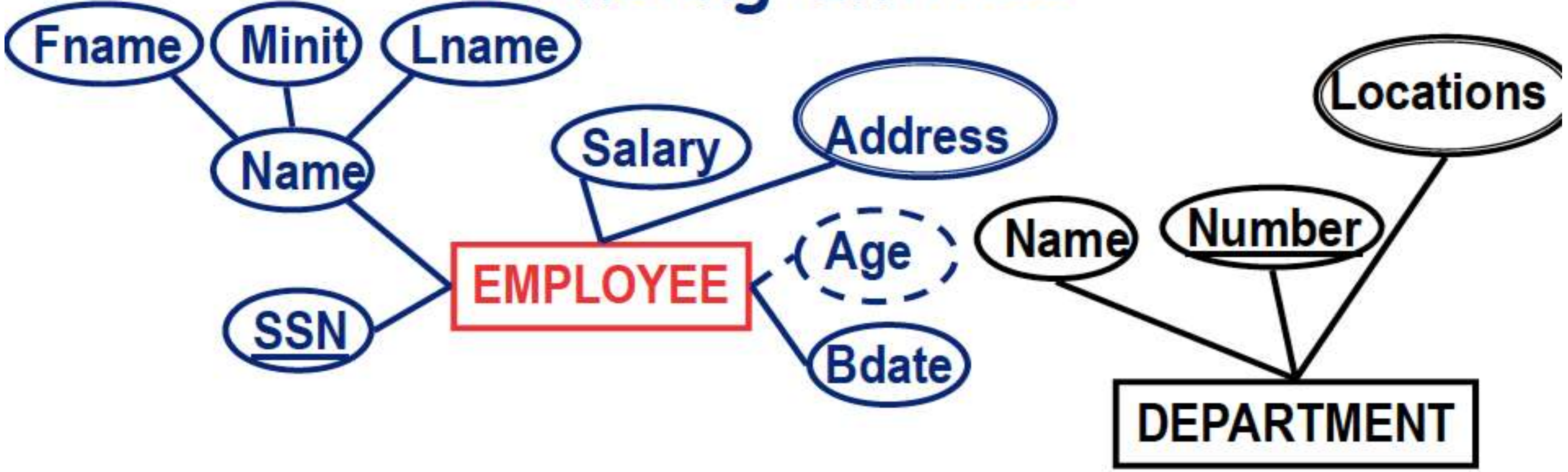
ER-to-Relational Mapping

3 For each weak entity type W in the ER schema with owner entity type E, create a relation R, and include all simple attributes of W as attributes of R, plus the primary key of E.

4 For each multivalued attribute A, create a new relation R that includes an attribute corresponding to A plus the primary key attribute K (as a foreign key in R) of the relation that represents the entity type or relationship type that has A as an attribute.

5 For each n-ary relationship type n >1, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types. Also include any simple attributes of the n-ary relationship type as attributes of S.

Handling Entities



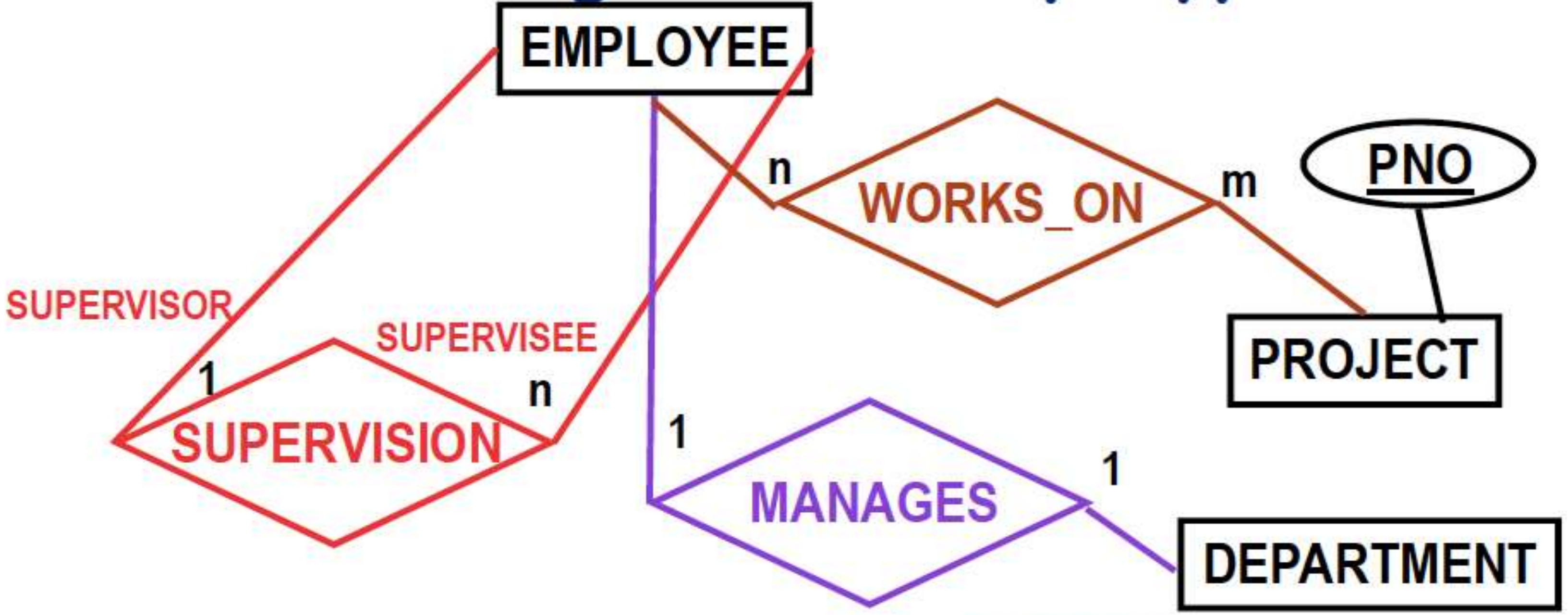
EMPLOYEE (<u>SSN</u>, FNAME, MI, LNAME, SALARY, Bdate)

EMP-ADD (<u>SSN,ADDRESS</u>)

DEPARTMENT(NAME,<u>NUMBER</u>)

DEPT_LOCNS(<u>NUMBER</u>,<u>LOCATION</u>)

Handling Relationship Types

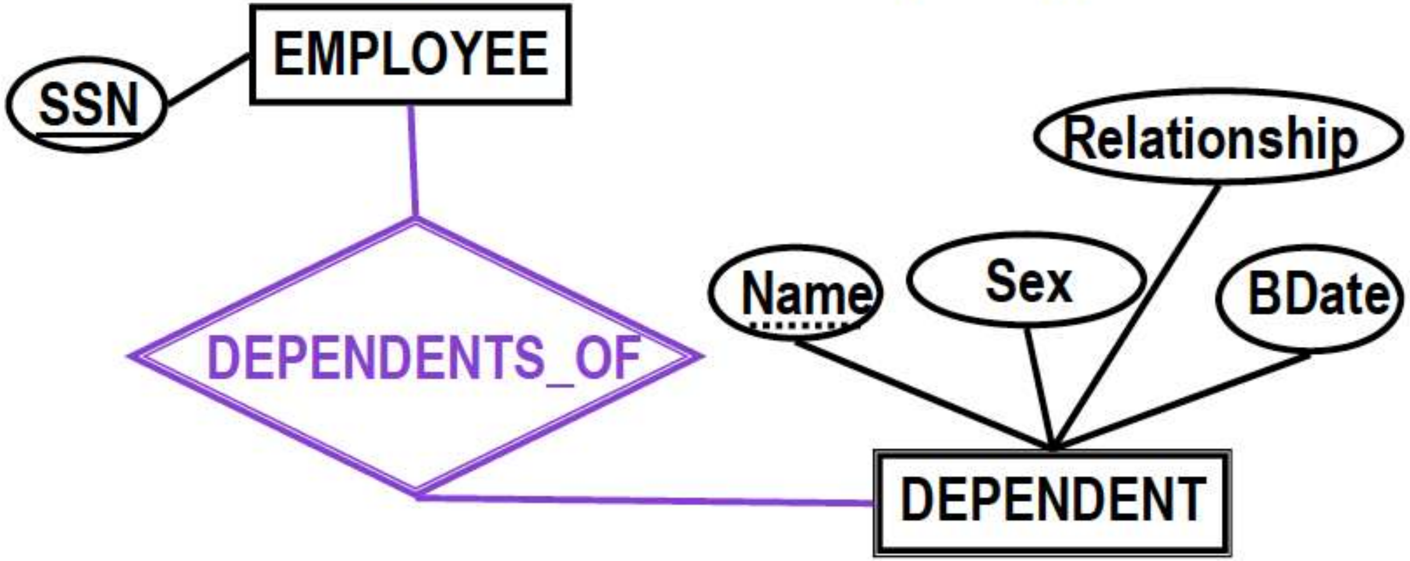


EMPLOYEE (<u>SSN</u>, FNAME, MI, LNAME, SUPSSN, SALARY, Bdate)

DEPARTMENT(NAME,<u>NUMBER</u>, MGRSSN)

WORKS_ON(<u>ESSN, PNO</u>)

Handling Weak Entity Types



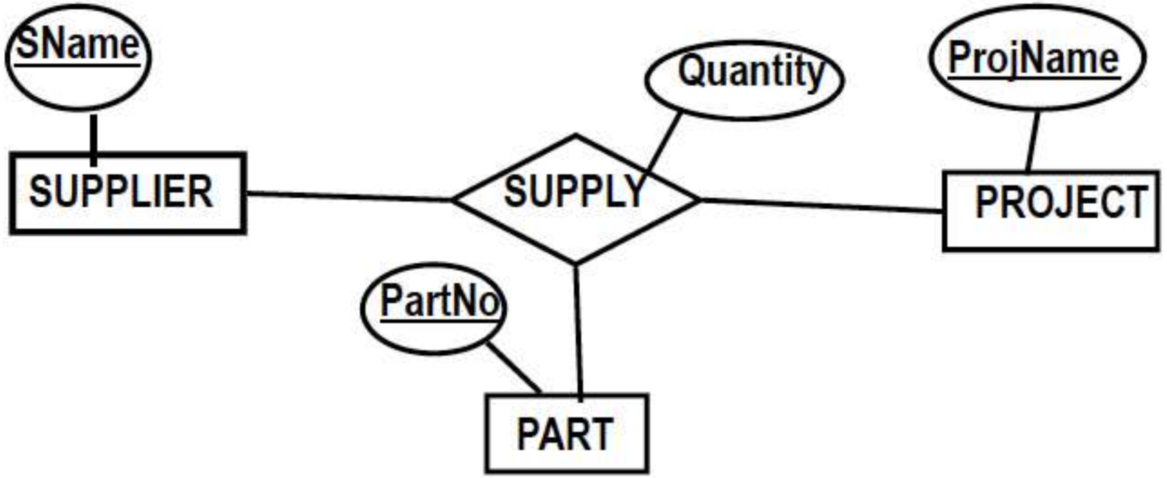
EMPLOYEE (<u>SSN</u>, FNAME, MI, LNAME, SUPSSN, SALARY, Bdate)

DEPARTMENT(NAME,<u>NUMBER</u>, MGRSSN)

WORKS_ON(<u>SSN, PNO</u>)

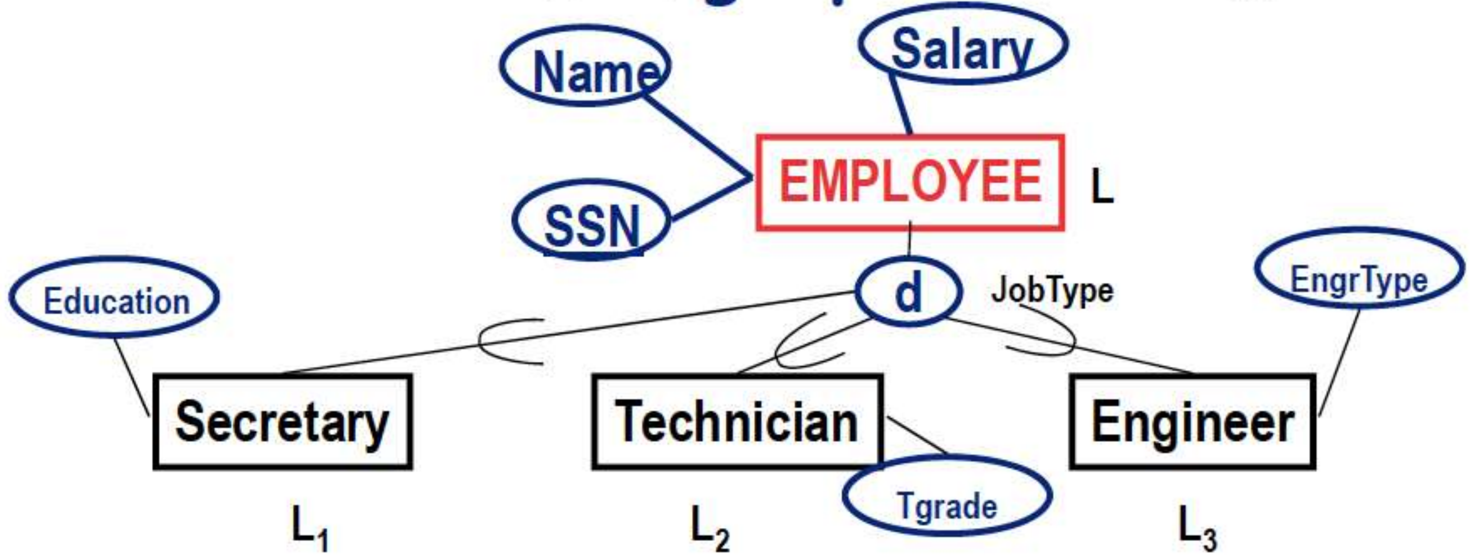
DEPENDENTS(<u>SSN,Name</u>,Sex,Relationship,Bdate)

Handling Relationships with higher Degree



SUPPLY(<u>Sname,ProjName,PartNo</u>,Quantity)

Handling Specialization



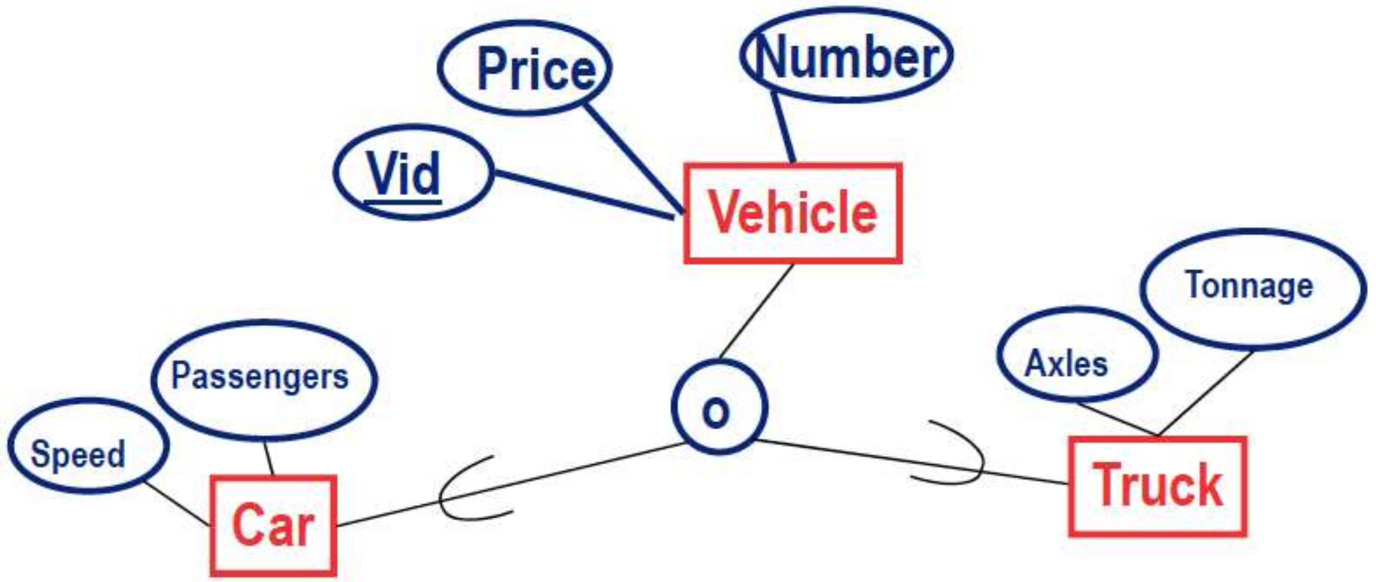
A) Employee(SSN,Name,Salary,JobType);Secretary(SSN,Education); Technician(SSN,Tgrade);Engineer(SSN,EngrType)

B) Secretary(SSN, Name,Salary,Education);Technician(SSN, Name,Salary,Tgrade); Engineer(SSN, Name,Salary, EngrType)

C) Employee(SSN,Name,Salary,JobType, Education,Tgrade,EngrType)

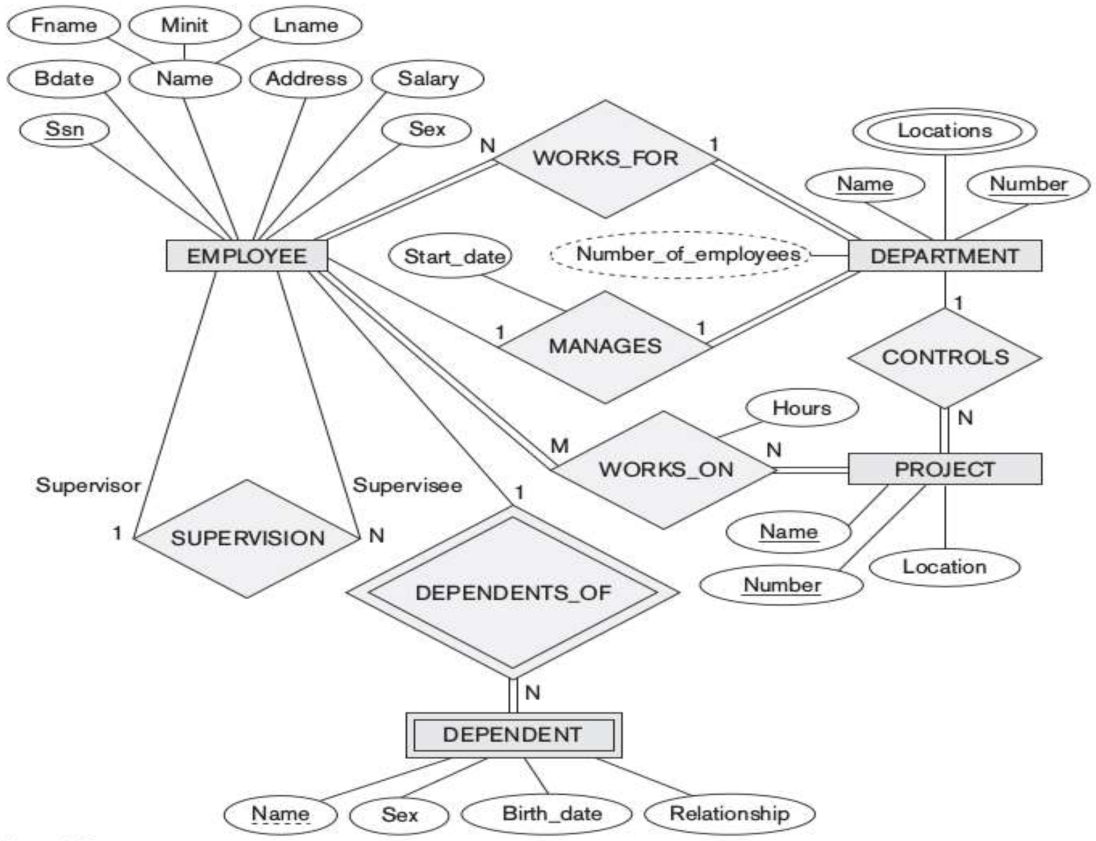
D) Employee(SSN,Name,Salary,Sflag, Education,Tflag,Tgrade, Eflag,EngrType)

Handling Generalization



Multi-purpose vehicles that can be used both as truck or car.

Vehicle(Vid, Price, Number, Cflag, Speed, Passengers, Tflag, Axles, Tonnage)


Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
Hours
N
M
N
WORKS_ON
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
DEPENDENTS_OF
Location
Number
N
DEPENDENT
Name
Sex
Birth_date
Relationship

COMPANY DATABASE

EMPLOYEE
FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO

DEPARTMENT
DNAME, DNUMBER, MGRSSN, MGRSTARTDATE

DEPT_LOCATIONS
DNUMBER, DLOCATION

PROJECT
PNAME, PNUMBER, PLOCATION, DNUM

WORKS_ON
ESSN, PNO, HOURS

DEPENDENT
ESSN, DEPENDENT_NAME, SEC, BDATE, RELATIONSHIP

Not in Syllabus of Tierce 2 beyond this point.

Relational Algebra

Relational Algebra is a collection of operations that are used to manipulate entire relations.

The result of each operation is a new relation, which can be further manipulated.

Relational Algebra Operators are divided into two groups:

- Set Operations like UNION, INTERSECTION, DIFFERENCE, and CARTESIAN PRODUCT.
- Relational Database Specific Operations like SELECT, PROJECT, and JOIN.

SELECT OPERATION (σ)

Selects the tuples (rows) from a relation R that satisfy a certain selection condition c

- Form of the operation $\sigma_c(R)$
- The condition c is an arbitrary Boolean expression on the attributes of R
- Resulting relation has the same set of attributes as R
- Resulting relation includes each tuple in r(R) whose attribute values satisfy the condition c

Examples

$\sigma_{DNO=4}$(Employee)

$\sigma_{SALARY>3000}$(Employee)

$\sigma_{(DNO=4 \text{ AND } SALARY>25000) \text{ OR } (DNO=5)}$ (Employee)

PROJECT Operation (Π)

Keeps only certain attributes (columns) from a relation R specified in an attribute list L

Form of operation: $\Pi_L(R)$

Resulting relation has only those attributes of R specified in L

Example:

$\Pi_{FNAME,\ LNAME,\ SALARY}$ (EMPLOYEE)

The project operation eliminates duplicate tuples in the resulting relation so that the result remains a mathematical set.

$\Pi_{SEX,SALARY}$(EMPLOYEE)

If several male employees have salary 3000, only a single tuple <M,3000> is kept in the resulting relation.

Sequences of operations

Several operations can be combined to form a relational algebra expression (query)

Example: Retrieve the names and salaries of employees who work in department 4:

a) $\Pi_{\text{FNAME, LNAME, SALARY}}(\sigma_{\text{DNO=4}}(\text{Employee}))$

Alternatively, intermediate relations for each step are specified

b) Dept4_Emps <-- $\sigma_{\text{DNO=4}}(\text{Employee})$

Result <-- $\Pi_{\text{FNAME, LNAME, SALARY}}(\text{Dept4\_Emps})$

Attributes can be renamed in the resulting relation:

Dept4_Emps <-- $\sigma_{\text{DNO=4}}(\text{Employee})$

R(FN,LN,Sal)<-- $\Pi_{\text{FNAME,LNAME,SALARY}}(\text{Dept4\_Emps})$