Designing Tables for an Oracle Database System

Database Course, Fall 2003

1

Add to Your .cshrc

- Add the following to your .cshrc file:

  source ~db/oraenv
- You will be able to use Oracle after you log out and log in again (or source .cshrc)
- You can run Oracle from the gx-es, pita, inferno, etc. Cannot run from xil-es
- If you are on xil, do rlogin or xon to one of these computers (e.g., rlogin gx-03)

2

Connecting to the Database

At the command line prompt, write:

sqlplus login/password@stud.cs

In the beginning your password is the same as your login. You can change your password with the command:

password

To disconnect, type: quit

Remember: Every command must end with a semicolon (;)

3

Running Commands from an .sql File

- Instead of typing commands into the SQLPLUS terminal, you can load commands from a file
  - Use the command @file from SQLPLUS to load the file *file.sql*
  - Invoke the SQLPLUS command with the extra parameter @*file* to load the file at connection:

sqlplus login/password@stud.cs @*file*

4

Spooling the Output

- Output can be placed in a file:
  - spool myFile.out
- Spooling can be turned off with:
  - spool off

5

Creating Tables

6

Creating a Table

The *basic* format of the CREATE TABLE command is:

```
CREATE TABLE TableName(
  Column1 DataType1 ColConstraint, ...
  ColumnN DataTypeN ColConstraint,
  TableConstraint1, ...
  TableConstraintM
);
```

7

An Example

```
CREATE TABLE Employee(
      SSN       NUMBER NOT NULL,
      Fname     VARCHAR2(20),
      Lname     VARCHAR2(20),
      Gender    CHAR(1),
      Salary    NUMBER(5) NOT NULL,
      Dept      NUMBER
);
```

8

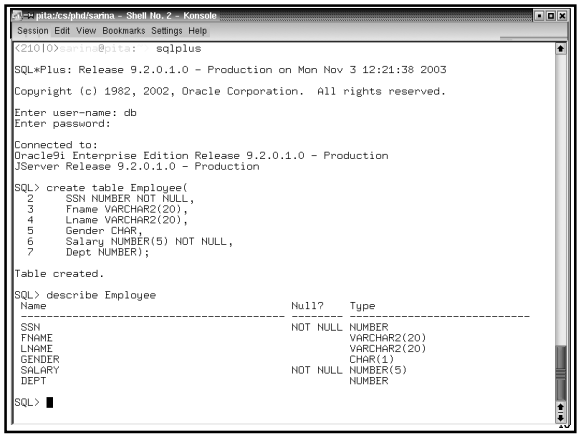
An Example (cont.)

Oracle is case insensitive in Column names!

If you type describe Employee you get:

Name	Null?	Type
SSN	NOT NULL	NUMBER
FNAME		VARCHAR2(20)
LNAME		VARCHAR2(20)
GENDER		CHAR(1)
SALARY	NOT NULL	NUMBER(5)
DEPT		NUMBER

9



10

Examples of Data Types

CHAR(*n*)	String of length *n* (*n* <= 2000)
VARCHAR2(*n*)	Variable length string of size <= *n* (*n* <= 4000)
LONG	Variable length string of length (<= 2GB)
CLOB	Character large object (<= 4GB)
BLOB	Binary large object (<= 4GB)
DATE	Valid dates (up to seconds)
TIMESTAMP	Valid timestamps (up to milliseconds)
NUMBER	Up to 40 digits
NUMBER(*n*)	Whole Number of size *n*
NUMBER(*n*,*m*)	Number of size *n* with *m* digits after decimal place
Others	XML, Abstract types, etc.

11

Example

- What happens if we insert:
  - 'abc' into char(5)?
  - 'abc' into varchar(5)?
  - 'abc' into char(2)?
  - 'abc' into varchar(2)?
  - 105.32 into number(3,2)?
  - 105.32 into number(5,2)?
  - 105.32 into number(4,1)?
  - 105.32 into number(3)?
  - 105.32 into number?
- Why not always use number and not number(n,m)?
- Why not always use varchar2(4000) or long?
- Where is the boolean datatype?

12

Constraints in Create Table

- Adding constraints to a table enables the database system to enforce data integrity.
- However, adding constraints also makes inserting data slower.
- Different types of constraints:
  - * Not Null
  - * Default Values
  - * Unique
  - * Primary Key
  - * Foreign Key
  - * Check Condition

13

Not Null Constraint

```
CREATE TABLE Employee(
        SSN         NUMBER NOT NULL,
        Fname       VARCHAR2(20),
        Lname       VARCHAR2(20),
        Gender      CHAR(1),
        Salary      NUMBER(5) NOT NULL,
        Dept        NUMBER
);
```

14

Default Values

```
CREATE TABLE Employee(
        SSN         NUMBER NOT NULL,
        Fname       VARCHAR2(20),
        Lname       VARCHAR2(20),
        Gender      CHAR(1) DEFAULT('F'),
        Salary      NUMBER(5) NOT NULL,
        Dept        NUMBER
);
```

15

Unique Constraint (Syntax 1)

```
CREATE TABLE Employee(
        SSN         NUMBER UNIQUE NOT NULL,
        Fname       VARCHAR2(20),
        Lname       VARCHAR2(20),
        Gender      CHAR(1) DEFAULT('F'),
        Salary      NUMBER(5) NOT NULL,
        Dept        NUMBER
);
```

16

Unique Constraint (Syntax 2)

```
CREATE TABLE Employee(
        SSN         NUMBER NOT NULL,
        Fname       VARCHAR2(20),
        Lname       VARCHAR2(20),
        Gender      CHAR(1) DEFAULT('F'),
        Salary      NUMBER(5) NOT NULL,
        Dept        NUMBER,
        UNIQUE(SSN)
);
```

17

Unique Constraint (Syntax 3)

```
CREATE TABLE Employee(
        SSN         NUMBER NOT NULL,
        Fname       VARCHAR2(20),
        Lname       VARCHAR2(20),
        Gender      CHAR(1) DEFAULT('F'),
        Salary      NUMBER(5) NOT NULL,
        Dept        NUMBER,
        constraint SSN_UN_CONS UNIQUE(SSN)
);
```

18

Unique Constraint (Another Example)

```
CREATE TABLE Employee(
      SSN         NUMBER UNIQUE NOT NULL,
      Fname       VARCHAR2(20),
      Lname       VARCHAR2(20),
      Gender      CHAR(1) DEFAULT('F'),
      Salary      NUMBER(5) NOT NULL,
      Dept        NUMBER,
      UNIQUE(Fname, Lname)
);
```

How else can this be written?

19

Primary Key Constraint

```
CREATE TABLE Employee(
      SSN         NUMBER PRIMARY KEY,
      Fname       VARCHAR2(20),
      Lname       VARCHAR2(20),
      Gender      CHAR(1) DEFAULT('F'),
      Salary      NUMBER(5) NOT NULL,
      Dept        NUMBER,
      UNIQUE(Fname, Lname)
);
```

Primary Key implies: * NOT NULL * UNIQUE.
There can only be one primary key.

20

Primary Key Constraint (Syntax 2)

```
CREATE TABLE Employee(
      SSN         NUMBER,
      Fname       VARCHAR2(20),
      Lname       VARCHAR2(20),
      Gender      CHAR(1) DEFAULT('F'),
      Salary      NUMBER(5) NOT NULL,
      Dept        NUMBER,
      UNIQUE(Fname, Lname),
      PRIMARY KEY(ssn)
);
```

What is Syntax 3?

21

Another Table

```
CREATE TABLE Department(
      Dept        NUMBER PRIMARY KEY,
      Name        VARCHAR2(20),
      ManagerId   NUMBER
);
```

? Shouldn t all department numbers in **Employee** appear in **Department**?

22

Foreign Key Constraint

```
CREATE TABLE Employee(
      SSN         NUMBER PRIMARY KEY,
      Fname       VARCHAR2(20),
      Lname       VARCHAR2(20),
      Gender      CHAR(1) DEFAULT('F'),
      Salary      NUMBER(5) NOT NULL,
      Dept        NUMBER,
      UNIQUE(Fname, Lname),
      FOREIGN KEY (Dept) REFERENCES
                        Department(Dept)
);
```

NOTE: Dept must be unique (or primary key) in Department

23

Foreign Key Constraint (Syntax 2)

```
CREATE TABLE Employee(
      SSN         NUMBER PRIMARY KEY,
      Fname       VARCHAR2(20),
      Lname       VARCHAR2(20),
      Gender      CHAR(1) DEFAULT('F'),
      Salary      NUMBER(5) NOT NULL,
      Dept        NUMBER,
      UNIQUE(Fname, Lname),
      FOREIGN KEY (Dept) REFERENCES
                        Department
);
```

NOTE: Dept must be the name of the field in Department, too

24

Understanding Foreign Keys

- The constraint on the last table should be read as: "The field **Dept** in **Employee** is a foreign key that references the field **Dept** in **Department**"
- Meaning: Every non-null value in the field **Dept** of **Employee** must appear in the field **Dept** of **Department**.

What happens to **Employee**s in department 312 when Department 312 is removed from the **Department** table?

25

Deleting a Referenced Value

- If nothing additional is specified, then Oracle will not allow Department 312 to be deleted if there are Employees working in this department.
- If the constraint is written as

  FOREIGN KEY (Dept) REFERENCES Department ON DELETE CASCADE

  then Employees working in 312 will be deleted automatically from the Employee table, when 312 is deleted from Departments

26

Cyclic Foreign Keys

We should revise the Department table:

```
CREATE TABLE Department(
      Dept          NUMBER PRIMARY KEY,
      Name          VARCHAR2(20),
      ManagerId     NUMBER,
      FOREIGN KEY (ManagerId)
         REFERENCES Employee(SSN)
);
```

Do you see a problem in defining these tables and in inserting data now?

27

Solution to Cyclic Constraints

Add one of the constraints later on (after insertion):

```
CREATE TABLE Department(
      Dept          NUMBER PRIMARY KEY,
      Name          VARCHAR2(20),
      ManagerId     NUMBER);
Insert data here…
ALTER TABLE Department
      ADD(FOREIGN KEY (ManagerId)
         REFERENCES Employee(SSN));
```

28

Check Conditions

- A check condition is a Boolean expression:
  - "And"s and "Or"s of conditions of the type X > 5…
- On a column: it can refer only to the column
- On a table: it can refer only to multiple columns in the table

29

Check Constraints

```
CREATE TABLE Employee(
      SSN           NUMBER PRIMARY KEY,
      Fname         VARCHAR2(20),
      Lname         VARCHAR2(20),
      Gender        CHAR(1) DEFAULT('F')
            CHECK(Gender = 'F' or Gender = 'M') ,
      Salary        NUMBER(5) NOT NULL,
      CHECK (Gender = 'M' or Salary > 10000)
);
```

30

Deleting a Table

- To delete the table **Employee** :

  DROP TABLE Employee;

- Be careful about the order of dropping when there are foreign key constraints. Why?
- Can use:

  DROP TABLE Employee cascade constraints;

31

Translating ER-Diagrams to Table Definitions

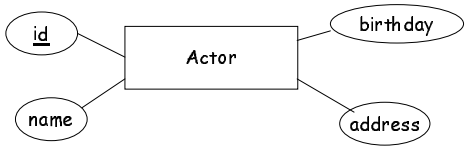
32

Relations vs. Tables

- We show how to translate ER-Diagrams to table definitions
- Sometimes, people translate ER-Diagrams to relation definition, which is more abstract than table definitions.
  - e.g., Employee(SSN, Fname, Lname, Gender, Salary, Dept);
  - table definitions contain, in addition, constraints and datatypes

33

Translating Entities

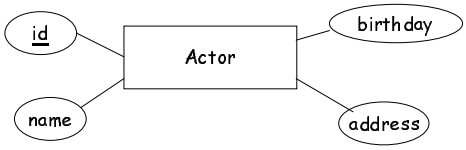


General Rule:

- Create a table with the name of the Entity.
- There is a column for each attribute
- The key in the diagram is the primary key of the table

34

Translating Entities

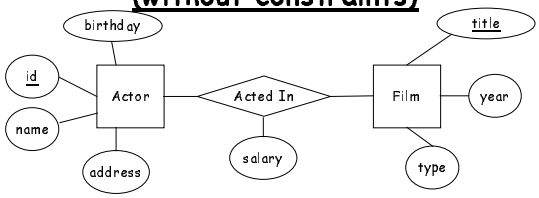


Relation: Actor (id, name, birthday, address)

```
create table Actor(id varchar2(20) primary key,
                   name varchar2(40),
                   birthday date,
                   address varchar2(100));
```

35

Translating Relationships (without constraints)

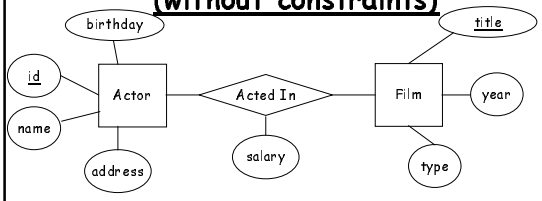


General Rule:

- Create a table with the name of the relationship
- The table has columns for all of the relationship's attributes and for the keys of each entity participating in the relationship
- What is the primary key of the table?
- What foreign keys are needed?

36

Translating relationships (without constraints)

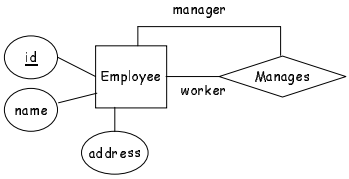


What would be the relation for ActedIn?

How would you define the table for ActedIn?

37

Translating Recursive Relationships (without constraints)

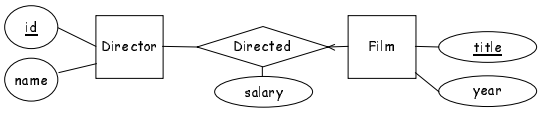


Relation: Actor (worker-id, manager-id)

What would be the table definition?

38

Translating relationships (key constraints): Option 1

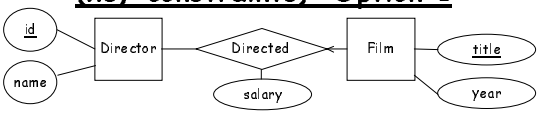


General Rule for Option 1:

- Same as without key constraints, except that the primary key is defined differently

39

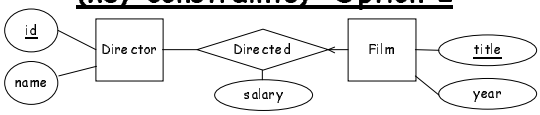
Translating relationships (key constraints): Option 1




```
create table Directed(
        id varchar2(20),
        title varchar2(40),
        salary integer,
        What primary and foreign keys are missing?
)
```

40

Translating relationships (key constraints): Option 2

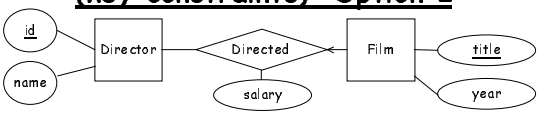


General Rule for Option 2:

- Do not create a table for the relationship
- Add information columns that would have been in the relationship's table to the table of the entity with the key constraint
- What is the disadvantage of this method?
- What is the advantage of this method?

41

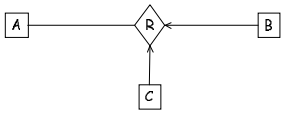
Translating relationships (key constraints): Option 2




```
create table Film(
        title varchar2(40),
        year integer,
        primary key (title),
        What 3 lines are missing?
)
```

42

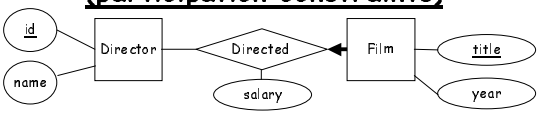
Translating relationships (key constraints)



- What are the different options for translating this diagram?

43

Translating relationships (participation constraints)

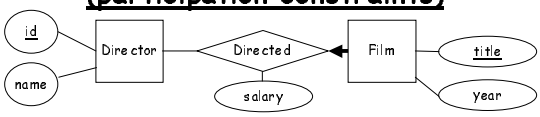


General Rule:

- If has both participation and key constraint, use Option 2 from before.
- Add the not null constraint to ensure that there will always be values for the key of the other entity

44

Translating relationships (participation constraints)

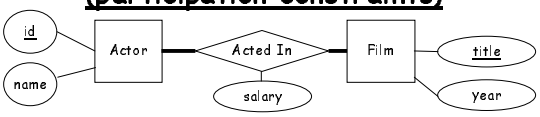



```
create table Film(
        title varchar2(40),
        year integer,
        id varchar2(20),
        salary integer,
        foreign key (id) references Director,
        primary key (title))
```

Where should we add NOT NULL?

45

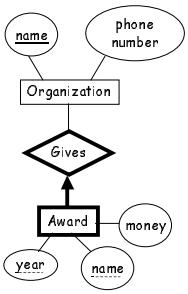
Translating relationships (participation constraints)



- How would we translate this?

46

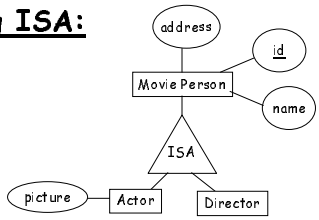
Translating Weak Entity Sets




```
create table award(
    name varchar2(40),
    year integer,
    money number(6,2),
    o_name varchar2(40),
    primary key(name, year, o_name),
    foreign key (o_name) references
        Organization(name)
        on delete cascade
)
```

47

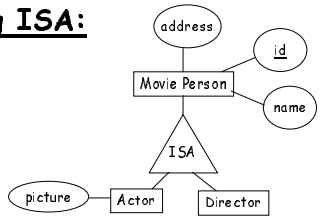
Translating ISA: Option 1




```
create table MoviePerson( ... )
create table Actor(id varchar2(20),
                   picture bfile,
                   primary key(id),
                   foreign key (id) references MoviePerson))
create table Director(...)
```

48

Translating ISA: Option 2



No table for MoviePerson!

```
create table Actor(id varchar2(20),
                   address varchar2(100),
                   name varchar2(20),
                   picture blob,
                   primary key(id));
create table Director(...)
```

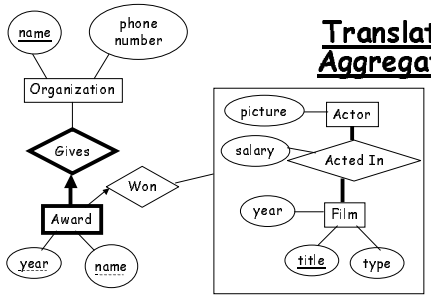
49

Which Option To Choose?

- What would you choose if:
  - Actor and Director DO NOT COVER MoviePerson?
  - Actor OVERLAPS Director?

50

Translating Aggregation



- Create table for Won using:
  - key of ActedIn
  - key of Award (careful, award is a weak entity)

51

Think About It

- Recall the ER-Diagram from last week
- What tables/relations would you define for the diagram?

52