

Table Alteration

1

Altering Tables

- Table definition can be altered after its creation
 - Adding columns
 - Changing columns definition
 - Dropping columns
 - Adding constraints
 - And more
- Use the reserved word **ALTER**

2

Altering Tables (continues)

- Adding a column:

```
ALTER TABLE Employee ADD (  
    Mname VARCHAR2(20),  
    Birthday DATE  
);
```

Cannot be
NOT NULL
unless the
table is empty

- Changing columns definition:

```
ALTER TABLE Employee Modify (  
    Mname VARCHAR2(10)  
);
```

3

Altering Tables (continues)

- Dropping columns:

```
ALTER TABLE Employee DROP COLUMN Mname;
```

Dropping multiple columns:

```
ALTER TABLE Employee DROP  
(Mname, Birthday);
```

- Adding constraints:

```
ALTER TABLE Department ADD(  
    FOREIGN KEY (ManagerId)  
    REFERENCES Employee(SSN));
```

4

User s Table List

- ORACLE may print tables that hold some general information about the tables in your database
- Such Tables are:
 - **Tab, Cat, User_Tables** (too detailed...)
- To see the list of all your tables you can print:
 - **SELECT * FROM Cat;**
 - **SELECT tname FROM Tab;**
 - **SELECT table_name from User_Tables;**

5

Table Data Maintenance

6

The Employee Table

> Describe Employee

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

7

Inserting a Row

- To insert a row into the Employee table:

```
INSERT INTO
  Employee(SSN, Fname, Lname, Salary)
VALUES(121, Sara , Cohen ,10000);
```

- The remaining columns get default values (or NULL)
- Order is not important
- When will this fail?

8

Some More Details

- The fields needn't be specified if values are specified for all columns and in the order defined by the table

- Example:**

```
INSERT INTO Employee
VALUES(121, Sara , Cohen , `F` , 10000);
```

9

Deleting Rows

- General format:

```
DELETE FROM Table WHERE Cond;
```

Deletes all rows satisfying *Cond* from *Table*
- For example, to remove the employee with SSN 121 from the Employee table:

```
DELETE FROM Employee WHERE
  SSN = 121;
```

10

Deleting Rows (continues)

- To remove all male employees having a salary greater than 15000 shekels:

```
DELETE FROM Employee WHERE Case sensitive
  Gender = M AND Salary > 15000;
```

- The **WHERE** clause is basically the same as one in a query

11

Updating Rows (continues)

- We can update fields of rows in a table
- General format:

```
UPDATE Table SET
  Field1=value1,,,FieldN=valueN
WHERE Cond
```
- Now we can reduce salaries instead of firing employees:

```
UPDATE Employee SET Salary = 15000
WHERE Gender = M AND
  Salary > 15000;
```

12

The ORACLE Bulk Loader

- A tool that provides easy insertion of large amounts of rows into tables.
- The idea: the field values of the rows are kept in a file, the format of which is defined by us.
- For example, it can automatically load 3 employees from the file myEmployees.dat that contains the following lines:

```
Sara|Cohen|121
Benny|Kimelfeld|134
Yaron|Kanza|156
```

13

The Control File

- The control file is the direct input of the loader
- A simple control file:

```
LOAD DATA
  INFILE <dataFile>
  [APPEND] INTO TABLE <tableName>
  FIELDS TERMINATED BY '<separator>'
  (<list of all attribute names to load>)
```

14

The Control File (continues)

- **<dataFile>**:
The name of the data file
- **<tableName>**:
The name of the table into which the data will be loaded (appended if APPEND is specified, or else the table must be empty)
- **<separator>**:
A string that separates two field values of a row
- The attributes are separated by commas and enclosed in parentheses

15

The Control File (continues)

- As an example, the following control file loads the employees from myEmployees.dat:

```
LOAD DATA
  INFILE myEmployees.dat
  INTO TABLE Employees
  FIELDS TERMINATED BY '|'
  (Fname, Lname, SSN)
```

- The attributes that are unspecified will be set to NULL

16

The Data File

- The Bulk Loader considers every single line to represent one row in the table
 - Even an empty line! (which will usually result in an error)
- Spaces are not ignored in the data file!
 - thus the rows sara|cohen|121 and sara|cohen|121 define different functionalities
- The NULL value is implied by the NULL keyword or the empty string

17

The Data File (continues)

- The control and the data files can be combined into one .ctl file using the following format:

```
LOAD DATA
  INFILE *
  INTO TABLE Employees
  FIELDS TERMINATED BY '|'
  (Fname, Lname, SSN)
BEGINDATA
  Sara|Cohen|121
  Benny|Kimelfeld|134
  Yaron|Kanza|156
```

18

The Bulk Invocation

- To invoke the bulk loader, issue the following command directly from the Unix shell:

```
sqlldr <yourName> control=<ctlFile>  
log=<logFile> bad=<badFile>
```

 - All fields are optional
 - File names that have no extension are automatically extended (by .dat, .log or .bad)
- Erroneous lines in the data file are ignored and written into **badFile**, and any other relevant information is written into **logFile**.

19

Bulk Loader Important Remarks

- Before using the Bulk Loader, make sure your personal ORACLE environment is properly set up
- The tables you fill using the Bulk Loader should be created prior to the loader invocation
- Before invoking the Bulk Loader you have to make sure that
NO SQLPLUS SESSIONS ARE OPEN!

20