Introduction to Data Definition Language in Oracle

11

HR Application Scenario
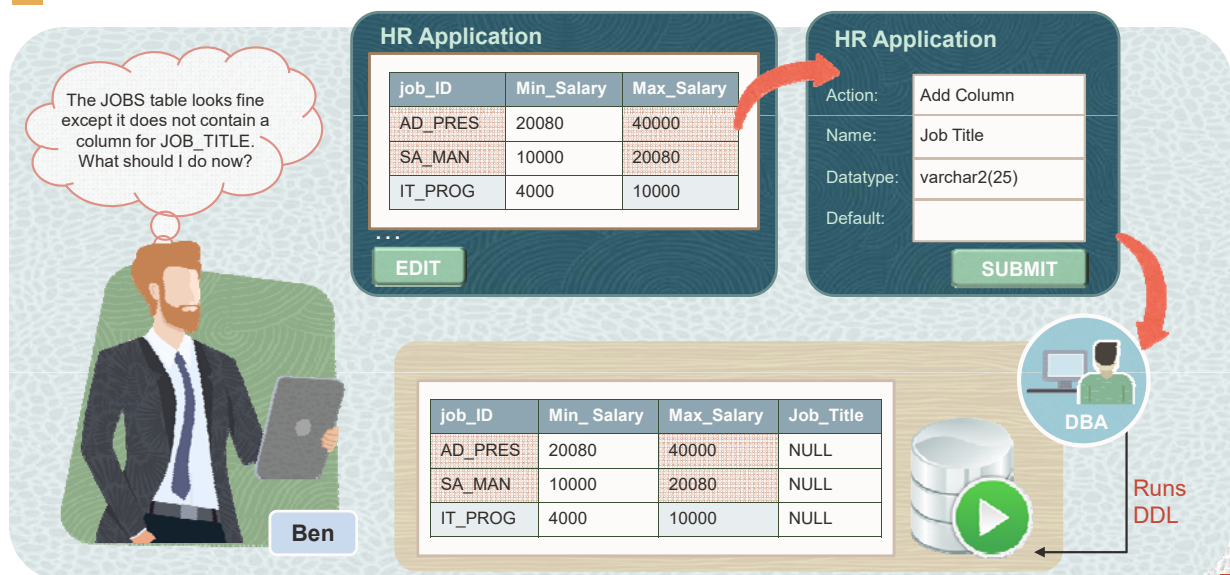
The JOBS table looks fine except it does not contain a column for JOB_TITLE. What should I do now?

HR Application

job_ID	Min_Salary	Max_Salary
AD_PRES	20080	40000
SA_MAN	10000	20080
IT_PROG	4000	10000

EDIT

HR Application

Action:	Add Column
Name:	Job Title
Datatype:	varchar2(25)
Default:	

SUBMIT

Ben

| job_ID	Min_ Salary	Max_Salary	Job_Title
AD_PRES	20080	40000	NULL
SA_MAN	10000	20080	NULL
IT_PROG	4000	10000	NULL

DBA

Runs DDL

Lesson Agenda

- Database objects
- CREATE TABLE statement
- Data types
- Overview of constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK constraints
- Creating a table using a subquery
- ALTER TABLE statement
- DROP TABLE statement

3

Database Objects

| Object	Description
Table	Is the basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative name to an object
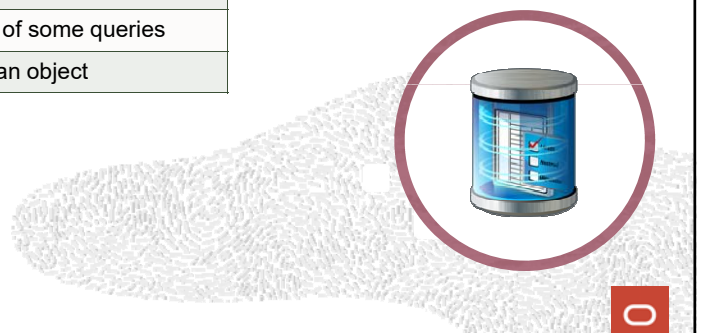
4

Naming Rules for Tables and Columns

Ensure that the table names and column names:

- Begin with a letter
- Are 128 characters long
- Contain only A–Z, a–z, 0–9, _, $, and #
- Do not duplicate the name of another object owned by the same user
- Are not Oracle server–reserved words
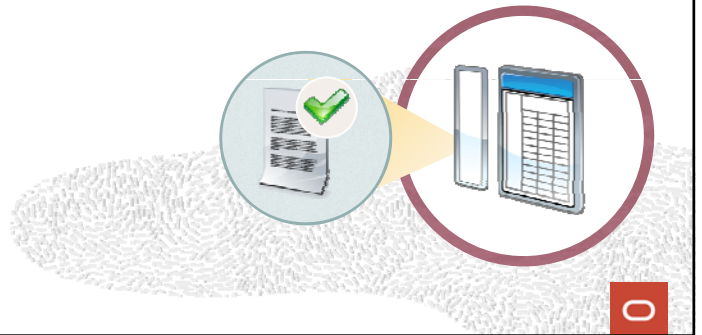
5

Lesson Agenda

- Database objects
- `CREATE TABLE` statement
- Data types
- Overview of constraints: `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK` constraints
- Creating a table using a subquery
- `ALTER TABLE` statement
- `DROP TABLE` statement

6

CREATE TABLE Statement

- You must have:
    - The CREATE TABLE privilege
    - A storage area

```
CREATE TABLE [schema.]table
        (column datatype [DEFAULT expr][, ...]);
```

- You specify:
    - The table name
    - The column name, column data type, and column size

Creating Tables

- Create the table:

```
CREATE TABLE dept
        (deptno       NUMBER(2),
         dname        VARCHAR2(14),
         loc          VARCHAR2(13),
         create_date DATE DEFAULT SYSDATE);
table DEPT created.
```

- Confirm table creation:

```
DESCRIBE dept
```

```
DESCRIBE dept
Name         Null Type
------------ ---- ------------
DEPTNO            NUMBER(2)
DNAME             VARCHAR2(14)
LOC               VARCHAR2(13)
CREATE_DATE       DATE
```

Lesson Agenda

- Database objects

- `CREATE TABLE` statement

- Data types

- Overview of constraints: `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK` constraints

- Creating a table using a subquery

- `ALTER TABLE` statement

- `DROP TABLE` statement

9

Data Types

Data Type	Description
`VARCHAR2(size)`	Variable-length character data
`CHAR(size)`	Fixed-length character data
`NUMBER(p, s)`	Variable-length numeric data
`DATE`	Date and time values
`LONG`	Variable-length character data (up to 2 GB)
`CLOB`	Maximum size is (4 gigabytes - 1) * (`DB_BLOCK_SIZE`).
`RAW and LONG RAW`	Raw binary data
`BLOB`	Maximum size is (4 gigabytes - 1) * (`DB_BLOCK_SIZE` initialization parameter (8 TB to 128 TB)).
`BFILE`	Binary data stored in an external file (up to 4 GB)
`ROWID`	A base-64 number system representing the unique address of a row in its table

DEFAULT Option

- Specify a default value for a column in the `CREATE TABLE` statement.

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- Literal values, expressions, or SQL functions are legal values.
- Another column's name or a pseudocolumn is an illegal value.
- The default data type must match the column data type.

```
CREATE TABLE hire_dates
        (id          NUMBER(8),
         hire_date DATE DEFAULT SYSDATE);
```

```
Table HIRE_DATES created.
```

Lesson Agenda

- Database objects
- `CREATE TABLE` statement
- Data types
- Overview of constraints: `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK` constraints
- Creating a table using a subquery
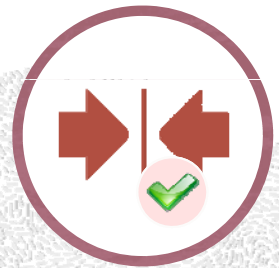- `ALTER TABLE` statement
- `DROP TABLE` statement

Including Constraints

- Constraints enforce rules at the table level.

- Constraints ensure consistency and integrity of the database.

- The following constraint types are valid:

  - NOT NULL

  - UNIQUE

  - PRIMARY KEY

  - FOREIGN KEY

  - CHECK

13

Defining Constraints

- Syntax:

```
CREATE TABLE [schema.]table
     (column datatype [DEFAULT expr]
     [column_constraint],
     ...
     [table_constraint][,...]);
```

- Column-level constraint syntax:

```
column [CONSTRAINT constraint_name] constraint_type,
```

- Table-level constraint syntax:

```
column,...
  [CONSTRAINT constraint_name] constraint_type
  (column, ...),
```

Defining Constraints: Example

- Example of a column-level constraint:

```
CREATE TABLE employees(
  employee_id  NUMBER(6)
    CONSTRAINT emp_emp_id_pk PRIMARY KEY,
  first_name   VARCHAR2(20),
  ...);
```

- Example of a table-level constraint:

```
CREATE TABLE employees(
  employee_id  NUMBER(6),
  first_name   VARCHAR2(20),
  ...
  job_id       VARCHAR2(10) NOT NULL,
  CONSTRAINT emp_emp_id_pk
    PRIMARY KEY (EMPLOYEE_ID));
```

15

NOT NULL Constraint

Ensures that null values are not permitted for the column:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	24000	(null)	90	SKING	515.123.4567	17-JUN-87
101	Neena	Kochhar	17000	(null)	90	NKOCHHAR	515.123.4568	21-SEP-89
102	Lex	De Haan	17000	(null)	90	LDEHAAN	515.123.4569	13-JAN-93
103	Alexander	Hunold	9000	(null)	60	AHUNOLD	590.423.4567	03-JAN-90
104	Bruce	Ernst	6000	(null)	60	BERNST	590.423.4568	21-MAY-91
107	Diana	Lorentz	4200	(null)	60	DLORENTZ	590.423.5567	07-FEB-99
124	Kevin	Mourgos	5800	(null)	50	KMOURGOS	650.123.5234	16-NOV-99
141	Trenna	Rajs	3500	(null)	50	TRAJS	650.121.8009	17-OCT-95
142	Curtis	Davies	3100	(null)	50	CDAVIES	650.121.2994	29-JAN-97
143	Randall	Matos	2600	(null)	50	RMATOS	650.121.2874	15-MAR-98
144	Peter	Vargas	2500	(null)	50	PVARGAS	650.121.2004	09-JUL-98
149	Eleni	Zlotkey	10500	0.2	80	EZLOTKEY	011.44.1344.429018	29-JAN-00
174	Ellen	Abel	11000	0.3	80	EABEL	011.44.1644.429267	11-MAY-96
176	Jonathon	Taylor	8600	0.2	80	JTAYLOR	011.44.1644.429265	24-MAR-98
178	Kimberely	Grant	7000	0.15	(null)	KGRANT	011.44.1644.429263	24-MAY-99
200	Jennifer	Whalen	4400	(null)	10	JWHALEN	515.123.4444	17-SEP-87
201	Michael	Hartstein	13000	(null)	20	MHARTSTE	515.123.5555	17-FEB-96
202	Pat	Fay	6000	(null)	20	PFAY	603.123.6666	17-AUG-97
205	Shelley	Higgins	12000	(null)	110	SHIGGINS	515.123.8080	07-JUN-94
206	William	Gietz	8300	(null)	110	WGIETZ	515.123.8181	07-JUN-94

NOT NULL constraint
(Primary Key enforces
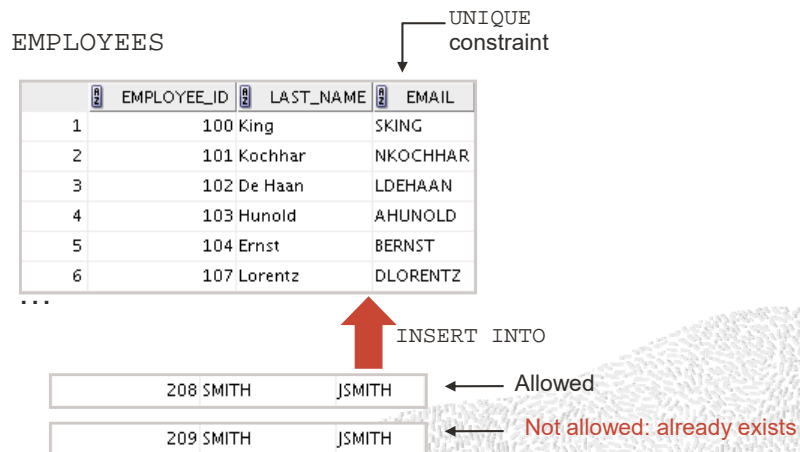NOT NULL constraint.)

NOT NULL
constraint

Absence of NOT NULL constraint (Any row
can contain a null value for this column.)

16

UNIQUE Constraint

EMPLOYEES

UNIQUE constraint

	EMPLOYEE_ID	LAST_NAME	EMAIL
1	100	King	SKING
2	101	Kochhar	NKOCHHAR
3	102	De Haan	LDEHAAN
4	103	Hunold	AHUNOLD
5	104	Ernst	BERNST
6	107	Lorentz	DLORENTZ

...

INSERT INTO

| 208 | SMITH | JSMITH | ← Allowed
| 209 | SMITH | JSMITH | ← Not allowed: already exists

17

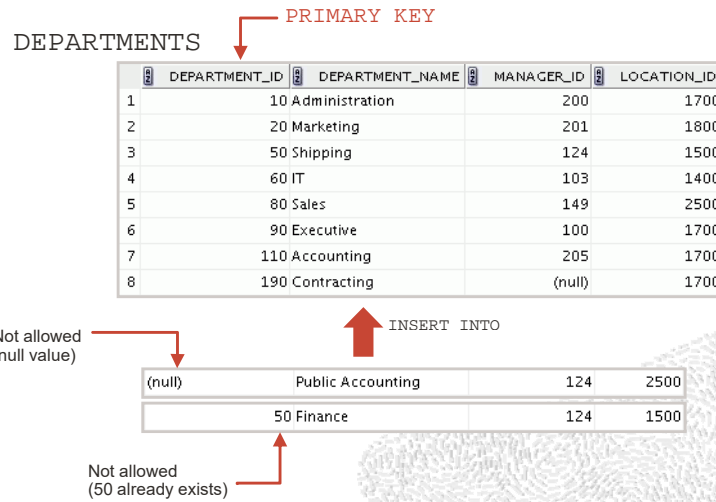# UNIQUE Constraint

Define at either the table level or the column level:

```
CREATE TABLE employees(
    employee_id        NUMBER(6),
    last_name          VARCHAR2(25) NOT NULL,
    email              VARCHAR2(25),
    salary             NUMBER(8,2),
    commission_pct     NUMBER(2,2),
    hire_date          DATE NOT NULL,
    ...
    CONSTRAINT emp_email_uk UNIQUE(email));
```
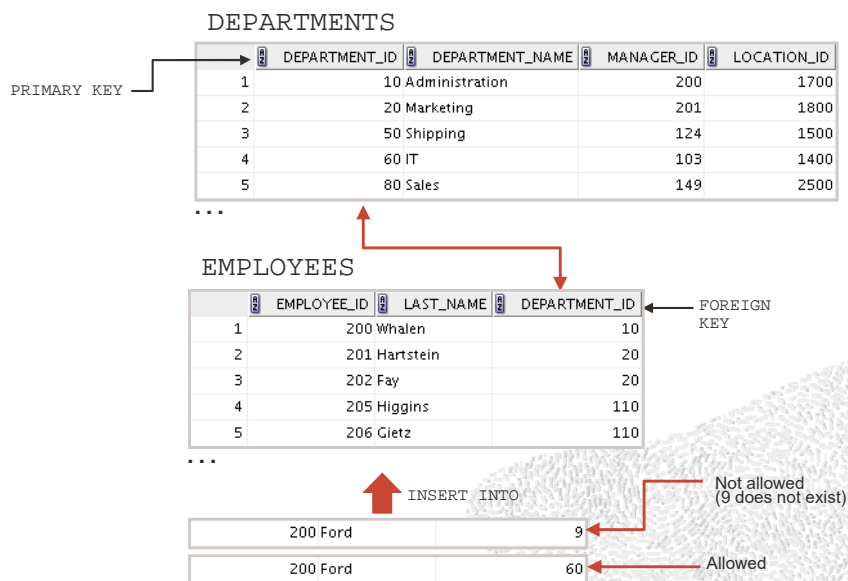
18

PRIMARY KEY Constraint

DEPARTMENTS

PRIMARY KEY

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500
6	90	Executive	100	1700
7	110	Accounting	205	1700
8	190	Contracting	(null)	1700

INSERT INTO

Not allowed
(null value)

(null)	Public Accounting	124	2500
50	Finance	124	1500

Not allowed
(50 already exists)

19

FOREIGN KEY Constraint

DEPARTMENTS

PRIMARY KEY

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500

...

EMPLOYEES

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	200	Whalen	10
2	201	Hartstein	20
3	202	Fay	20
4	205	Higgins	110
5	206	Gietz	110

FOREIGN KEY

...

INSERT INTO

200	Ford	9
200	Ford	60

Not allowed
(9 does not exist)

Allowed

20

FOREIGN KEY Constraint

Define at either the table level or the column level:

```
CREATE TABLE employees(
    employee_id      NUMBER(6),
    last_name        VARCHAR2(25) NOT NULL,
    email            VARCHAR2(25),
    salary           NUMBER(8,2),
    commission_pct   NUMBER(2,2),
    hire_date        DATE NOT NULL,
...
    department_id    NUMBER(4),
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
      REFERENCES departments(department_id),
    CONSTRAINT emp_email_uk UNIQUE(email));
```

21

CHECK Constraint

- Defines a condition that each row must satisfy

- Cannot reference columns from other tables

```
..., salary     NUMBER(2)
    CONSTRAINT emp_salary_min
          CHECK (salary > 0),...
```

CREATE TABLE: Example

```
CREATE TABLE teach_emp (
        empno       NUMBER(5) PRIMARY KEY,
        ename       VARCHAR2(15) NOT NULL,
        job         VARCHAR2(10),
        mgr         NUMBER(5),
        hiredate    DATE DEFAULT (sysdate),
        photo       BLOB,
        sal         NUMBER(7,2),
        deptno      NUMBER(3) NOT NULL
                     CONSTRAINT admin_dept_fkey
    REFERENCES
                  departments(department_id));
```

Violating Constraints

Department 55 does not exist.

```
UPDATE employees
SET     department_id = 55
WHERE   department_id = 110;
```

```
Error starting at line : 1 in command -
UPDATE employees
SET     department_id = 55
WHERE   department_id = 110
Error report -
ORA-02291: integrity constraint (ORA1.EMP_DEPT_FK) violated - parent key not found
```

Violating Constraints

You cannot delete a row that contains a primary key that is used as a foreign key in another table.

```
DELETE FROM departments
WHERE department_id = 60;
```

```
Error starting at line : 1 in command -
delete from departments
where department id = 60
Error report -
ORA-02292: integrity constraint (ORA1.EMP_DEPT_FK) violated - child record found
```

Lesson Agenda

- Database objects
- Data types
- CREATE TABLE statement
- Overview of constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK constraints
- Creating a table using a subquery
- ALTER TABLE statement
- DROP TABLE statement

Creating a Table Using a Subquery

- Create a table and insert rows by combining the `CREATE TABLE` statement and the `AS subquery` option.

```
CREATE TABLE table
        [(column, column...)]
AS subquery;
```

- Match the number of specified columns to the number of subquery columns.

- Define columns with column names and default values.

Creating a Table Using a Subquery

```
CREATE TABLE    dept80
  AS
    SELECT   employee_id, last_name,
             salary*12 ANNSAL,
             hire_date
    FROM     employees
    WHERE    department_id = 80;
```

```
Table DEPT80 created.
```

```
DESCRIBE dept80
```

```
Name                           Null     Type
------------------------------ -------- -------------
EMPLOYEE_ID                              NUMBER(6)
LAST_NAME                      NOT NULL VARCHAR2(25)
ANNSAL                                   NUMBER
HIRE_DATE                      NOT NULL DATE
```

Lesson Agenda

- Database objects

- Data types

- CREATE TABLE statement

- Overview of constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK constraints

- Creating a table using a subquery
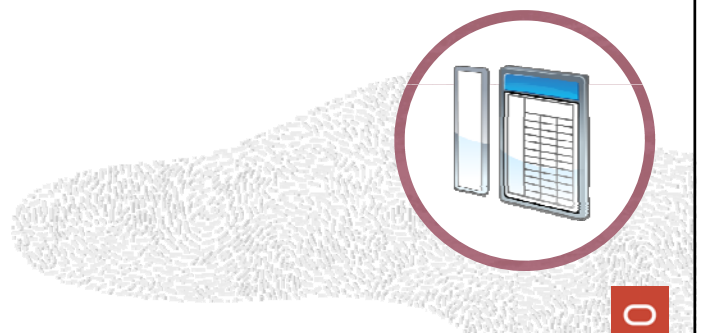
- ALTER TABLE statement

- DROP TABLE statement

29

ALTER TABLE Statement

Use the ALTER TABLE statement to:

- Add a new column

- Modify an existing column definition

- Define a default value for the new column

- Drop a column

- Rename a column

- Change table to read-only status

30

ALTER TABLE Statement

Use the `ALTER TABLE` statement to add, modify, or drop columns:

```
ALTER TABLE table
ADD         (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
MODIFY      (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
DROP (column [, column] …);
```

Adding a Column

- You use the `ADD` clause to add columns:

```
ALTER TABLE dept80
ADD             (job_id VARCHAR2(9));
```

Table DEPT80 altered.

- The new column becomes the last column:

	EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
1	149	Zlotkey	126000	29-JAN-16	(null)
2	174	Abel	132000	11-MAY-12	(null)
3	176	Taylor	103200	24-MAR-14	(null)
4	206	Gietz	99600	07-JUN-10	(null)

Modifying a Column

- You can change a column's data type, size, and default value.

```
ALTER TABLE dept80
MODIFY            (last_name VARCHAR2(30));
```
Table DEPT80 altered.

Size of the last_name
column is modified.

- A change to the default value of a column affects only subsequent insertions to the table.

33

Dropping a Column

Use the DROP COLUMN clause to drop columns that you no longer need from the table:

```
ALTER TABLE  dept80
DROP (job_id);
```
Table DEPT80 altered.

	EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE
1	149	Zlotkey	126000	29-JAN-16
2	174	Abel	132000	11-MAY-12
3	176	Taylor	103200	24-MAR-14
4	206	Gietz	99600	07-JUN-10

Read-Only Tables

You can use the `ALTER TABLE` syntax to:

- Put a table in read-only mode, which prevents DDL or DML changes during table maintenance

- Put the table back into read/write mode

```
ALTER TABLE employees READ ONLY;

-- perform table maintenance and then
-- return table back to read/write mode

ALTER TABLE employees READ WRITE;
```

35

Lesson Agenda

- Database objects

- Data types

- `CREATE TABLE` statement

- Overview of constraints: `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK` constraints

- Creating a table using a subquery

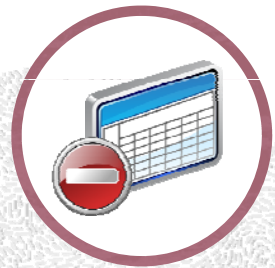- `ALTER TABLE` statement

- `DROP TABLE` statement

36

Dropping a Table

- Moves a table to the recycle bin

- Removes the table and all its data entirely if the `PURGE` clause is specified

- Invalidates dependent objects and removes object privileges on the table

```
DROP TABLE dept80;

Table DEPT80 dropped.
```

37

Summary

In this lesson, you should have learned how to use the `CREATE TABLE`, `ALTER TABLE`, and `DROP TABLE` statement to create a table, modify a table and columns, and include constraints.

- Categorize the main database objects

- Review the table structure

- List the data types that are available for columns

- Create a simple table

- Explain how constraints are created at the time of table creation