

The background is a light gray gradient. It is decorated with several realistic water droplets of various sizes, some with highlights and shadows, scattered across the frame. In the upper center, there is a faint, circular, embossed-style logo that appears to be a university crest or seal.

# RELATIONAL DATABASES

# OBJECTIVES

- DEFINE AND GIVE AN EXAMPLE OF A PRIMARY KEY, FOREIGN KEY AND CHECK CONSTRAINTS.
- EXPLAIN THE PURPOSE OF DEFINING PRIMARY KEY, FOREIGN KEY AND CHECK CONSTRAINTS.
- DEMONSTRATE THE CREATION OF THESE CONSTRAINTS AT COLUMN OR TABLE LEVEL.

# PRIMARY KEY CONSTRAINTS

- A PRIMARY KEY CONSTRAINT IS A RULE THAT THE VALUES IN ONE COLUMN OR COMBINATION OF COLUMNS MUST UNIQUELY IDENTIFY EACH ROW IN A TABLE.
- NO PRIMARY KEY VALUE CAN APPEAR IN MORE THAN ONE ROW IN A TABLE.
- TO SATISFY THE PRIMARY KEY CONSTRAINT BOTH OF THE FOLLOWING MUST BE TRUE:
  - NO COLUMN THAT IS PART OF THE PRIMARY KEY CAN CONTAIN A NULL.
  - A TABLE CAN ONLY HAVE ONE PRIMARY KEY.
- A PRIMARY KEY CONSTRAINTS CAN BE DEFINED AT THE COLUMN OR TABLE LEVEL.
- IF IT IS A COMPOSITE KEY IT MUST BE DEFINED AT THE TABLE LEVEL.
- IT IS COMMON PRACTICE TO USE \_PK FOR THE NAME

# PRIMARY KEY CONSTRAINTS

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
first_name VARCHAR2(14),
last_name VARCHAR2(13));
```

```
CREATE TABLE clients
(client_number NUMBER(4),
first_name VARCHAR2(14),
last_name VARCHAR2(13),
CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

# PRIMARY KEY CONSTRAINTS

- A COMPOSITE KEY IS DEFINED AT THE TABLE LEVEL AND ALL COLUMN NAMES ARE INCLUDED IN THE PARENTHESIS SEPARATED BY COMMAS.

```
CREATE TABLE copy_job_history  
(employee_id NUMBER(6,0),  
  start_date DATE,  
  job_id VARCHAR2(10),  
  department_id NUMBER(4,0),  
  CONSTRAINT copy_jhist_id_st_date_pk PRIMARY KEY(employee_id, start_date));
```

- WHEN DEFINING ANY CONSTRAINT AT THE TABLE LEVEL THE COLUMN NAME MUST EXIST UPON WHICH THE CONSTRAINT IS TO BE APPLIED.

# FOREIGN KEY (REFERENTIAL INTEGRITY) CONSTRAINTS

- FOREIGN KEY CONSTRAINTS ARE ALSO CALLED 'REFERENTIAL INTEGRITY' CONSTRAINTS.
- THEY DESIGNATE A COLUMN OR COMBINATION OF COLUMNS AS A FOREIGN KEY.
- THEY LINK BACK TO THE PRIMARY KEY IN ANOTHER TABLE.
- THE TABLE CONTAINING THE FOREIGN KEY IS CALLED THE 'CHILD' TABLE AND THE TABLE IT REFERENCES IS CALLED THE 'PARENT' TABLE.

# FOREIGN KEY (REFERENTIAL INTEGRITY) CONSTRAINTS

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

- TO SATISFY A REFERENTIAL INTEGRITY CONSTRAINT A FOREIGN KEY VALUE MUST MATCH AN EXISTING VALUE IN THE PARENT TABLE OR BE NULL
- A PRIMARY KEY VALUE CAN EXIST WITHOUT A CORRESPONDING FOREIGN KEY BUT NOT VISA VERSA.

# FOREIGN KEY (REFERENTIAL INTEGRITY) CONSTRAINTS

- BEFORE YOU DEFINE A REFERENTIAL INTEGRITY CONSTRAINT IN THE CHILD TABLE, THE REFERENCED PRIMARY KEY CONSTRAINT IN THE PARENT TABLE MUST ALREADY EXIST.
- IT IS GOOD PRACTICE TO USE \_FK FOR THE NAMING OF A FOREIGN KEY CONSTRAINT.

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0) CONSTRAINT c_emps_dept_id_fk
                                REFERENCES departments(department_id),
 email VARCHAR2(25));
```



# FOREIGN KEY (REFERENTIAL INTEGRITY) CONSTRAINTS

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
first_name VARCHAR2(20),
last_name VARCHAR2(25),
department_id NUMBER(4,0),
email VARCHAR2(25),
CONSTRAINT c_ems_dept_id_fk FOREIGN KEY (department_id)
REFERENCES departments(department_id));
```

# MAINTAINING REFERENTIAL INTEGRITY

- ON DELETE CASCADE ENABLES THE DEPENDENT ROWS IN THE CHILD TABLE TO BE DELETED WHEN A ROW IN THE PARENT TABLE IS DELETED.
- IF THE FOREIGN KEY DOES NOT HAVE AN ON DELETE CASCADE OPTION (DEFAULT), REFERENCED ROWS IN THE PARENT TABLE CANNOT BE DELETED.

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
first_name VARCHAR2(20),
last_name VARCHAR2(25),
department_id NUMBER(4,0),
email VARCHAR2(25),
CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
REFERENCES copy_departments(department_id) ON DELETE CASCADE);
```

# FOREIGN KEY (REFERENTIAL INTEGRITY) CONSTRAINTS

- RATHER THAN HAVING THE ROWS IN THE CHILD TABLE DELETED WHEN USING THE ON DELETE CASCADE OPTION, THE CHILD ROWS CAN BE FILLED WITH NULL VALUES
- ON DELETE SET NULL

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
first_name VARCHAR2(20),
last_name VARCHAR2(25),
department_id NUMBER(4,0),
email VARCHAR2(25),
CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
REFERENCES copy_departments(department_id) ON DELETE SET NULL);
```

# CHECK CONSTRAINTS

- THE CHECK CONSTRAINT EXPLICITLY DEFINES A CONDITION THAT MUST BE MET.
- TO SATISFY THE CONSTRAINT, EACH ROW IN THE TABLE MUST MAKE THE CONDITION EITHER TRUE OR UNKNOWN (DUE TO A NULL).
- THE CONDITION OF A CHECK CONSTRAINT CAN REFER TO ANY COLUMN IN THE SPECIFIED TABLE BUT NOT TO COLUMNS OF OTHER TABLES.
- IF A CHECK CONSTRAINT REFERENCES MORE THAN ONE COLUMN IT MUST BE DEFINED AT THE TABLE LEVEL.

# CHECK CONSTRAINTS

```
CREATE TABLE copy_job_history
(employee_id NUMBER(6,0),
 start_date DATE,
 end_date DATE,
 job_id VARCHAR2(10),
 department_id NUMBER(4,0),
 CONSTRAINT cjhist_emp_id_st_date_pk
          PRIMARY KEY(employee_id, start_date),
 CONSTRAINT cjhist_end_ck CHECK (end_date > start_date));
```

# CHECK CONSTRAINTS

- A CHECK CONSTRAINT MUST ONLY BE ON THE ROW WHERE THE CONSTRAINT IS DEFINED.
- A CHECK CONSTRAINT CANNOT BE USED IN QUERIES THAT REFER TO VALUES IN OTHER ROWS.
- A CHECK CONSTRAINT CANNOT CONTAIN CALLS TO THE FUNCTIONS SYSDATE, UID, USER, OR USERENV.
- A CHECK CONSTRAINT CANNOT USE THE PSEUDOCOLUMNS CURRVAL, NEXTVAL, LEVEL, OR ROWNUM.
- A SINGLE COLUMN CAN HAVE MULTIPLE CHECK CONSTRAINTS.
- THEY CAN BE DEFINED AT THE COLUMN OR TABLE LEVEL.

# CHECK CONSTRAINTS

– Column-level syntax:

```
salary NUMBER(8,2) CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```

– Table-level syntax:

```
CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```