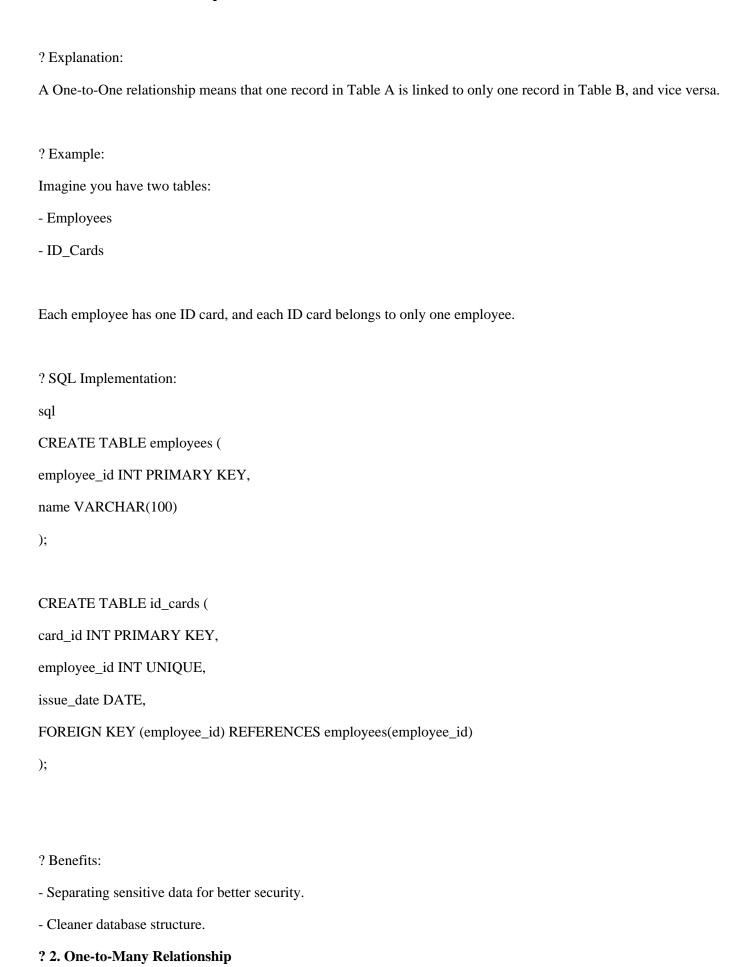
? 1. One-to-One Relationship



A One-to-Many relationship means one record in Table A is related to many records in Table B, but each record in Table B is related to only one record in Table A.
? Example:
Imagine two tables:
- Teachers
- Students
One teacher can have many students, but each student is assigned to one teacher only.
? SQL Implementation:
```sql
CREATE TABLE teachers (
teacher_id INT PRIMARY KEY,
name VARCHAR(100)
);
CREATE TABLE students (
student_id INT PRIMARY KEY,
name VARCHAR(100),
teacher_id INT,
FOREIGN KEY (teacher_id) REFERENCES teachers(teacher_id)
);
? Benefits:
- Efficient management of parent-child data structures.
- Easier data retrieval and analysis.
? 3. Many-to-Many Relationship

? Explanation:

? Explanation:

A *Many-to-Many* relationship means *multiple records* in *Table A* can relate to *multiple records* in *Table B*. You must use a *junction (intermediate) table* to manage this. ? Example: Imagine two tables: - `Students` - `Courses` Each student can enroll in many courses, and each course can have many students. ? SQL Implementation: sql CREATE TABLE students ( student_id INT PRIMARY KEY, name VARCHAR(100) ); CREATE TABLE courses ( course_id INT PRIMARY KEY, title VARCHAR(100) ); CREATE TABLE student_courses ( student_id INT, course_id INT, PRIMARY KEY (student_id, course_id), FOREIGN KEY (student_id) REFERENCES students(student_id),

- Allows flexible and powerful data linking.

? Benefits:

- Ideal for systems that handle enrollments, memberships, tags, etc.

FOREIGN KEY (course_id) REFERENCES courses(course_id)