

DATABASE SYSTEMS COURSE 2021/2022

SESSION 9

SESSION 9

- **RELATIONAL DATABASES BASICS**
- TABLES RELATIONSHIP
- NORMALIZATION
- RELATIONAL DATABASE CASE 1
- RELATIONAL DATABASE CASE 2
- RELATIONAL DATABASES CONSTRAINTS

Definition

- Cares the integrity of the data:
 - Completeness and consistency of the data
 - Safety of the data
- RDBMS manage constraints
 - All related table should have a primary key constraint.
 - All reference to other tables should be specified with a foreign key constraint.
- Any association between two entities is called relationship. There are three types of relationship:
 - One to One
 - One to Many or Many to One
 - Many to Many

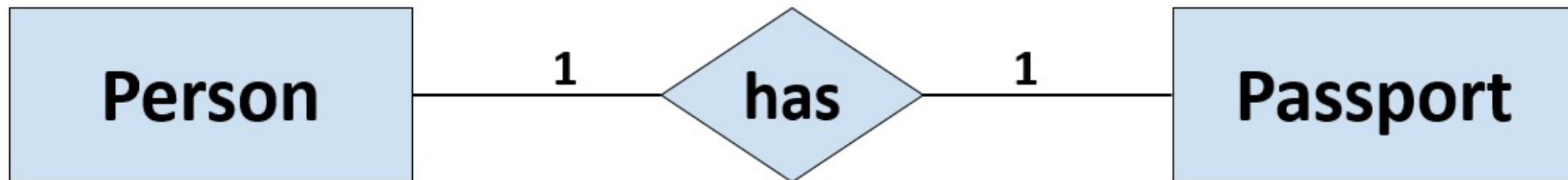
SESSION 9

- RELATIONAL DATABASES BASICS
- **TABLES RELATIONSHIP**
- NORMALIZATION
- RELATIONAL DATABASE CASE 1
- RELATIONAL DATABASE CASE 2
- RELATIONAL DATABASES CONSTRAINTS

TABLES RELATIONSHIP

One to one

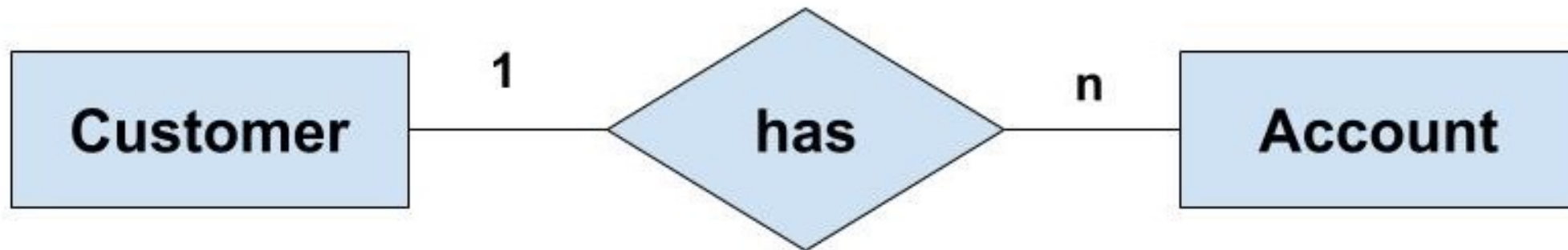
- Such a relationship exists when each record of one table is related to only one record of the other table.



TABLES RELATIONSHIP

One to Many/Many to One

- Such a relationship exists when each record of one table can be related to one or more than one record of the other table. This relationship is the most common relationship found.



TABLES RELATIONSHIP

One to Many/Many to One

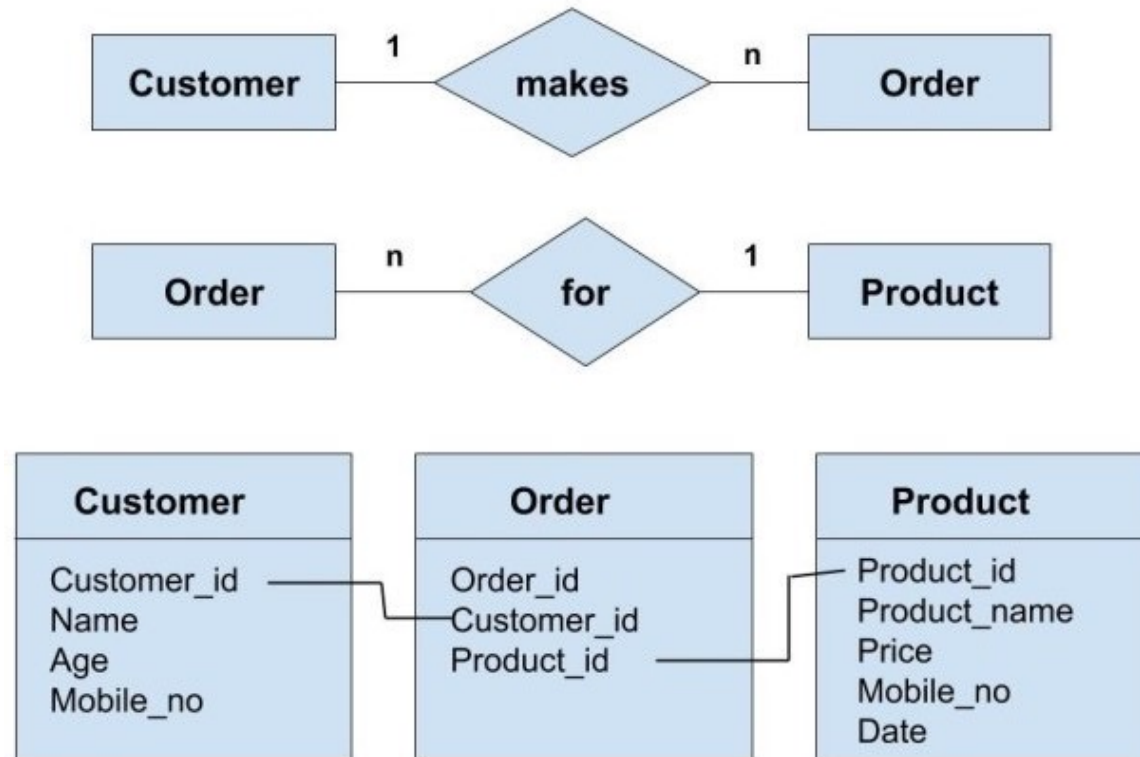
- A many-to-many relationship can be seen as a two one-to-many relationship which is linked by a 'linking table' or 'associate table'.



TABLES RELATIONSHIP

One to Many/Many to One (linking table)

- We can have the 'Order' entity as a linking table which links the 'Customer' and 'Product' entity. We can break this many-to-many relationship in two one-to-many relationships.



SESSION 9

- RELATIONAL DATABASES BASICS
- TABLES RELATIONSHIP
- **NORMALIZATION**
- RELATIONAL DATABASE CASE 1
- RELATIONAL DATABASE CASE 2
- RELATIONAL DATABASES CONSTRAINTS

Definition

- Normalization is a process of organizing the data in database to avoid data redundancy.
- It helps to achieve the integrity of the data.
- There are four normal forms
 - First normal form(1NF)
 - Second normal form(2NF)
 - Third normal form(3NF)
 - Boyce & Codd normal form (BCNF)

First Normal Form (1NF)

- As per the rule of first normal form:
 - An attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

mp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212 9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123 8123450987



emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212
102	Jon	Kanpur	9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123
104	Lester	Bangalore	8123450987



Carlos DIEZ dejanos
trabajar;)

Second Normal Form (2NF)

- A table is said to be in 2NF if both the following conditions hold:
 - Table is in 1NF (First normal form)
 - No non-prime attribute is dependent on the proper subset of any candidate key of table.

teacher_id	subject	teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40
333	Chemistry	40

Candidate Keys:

{teacher_id, subject}



teacher_id	teacher_age
111	38
222	38
333	40



teacher_id	subject
111	Maths
111	Physics
222	Biology
333	Physics
333	Chemistry

Third Normal Form (3NF)

- A table design is said to be in 3NF if both the following conditions hold:
 - Table must be in 2NF
 - Transitive functional dependency of non-prime attribute on any super key should be removed.

Project	ProjctTitle	Manager	Telf
132N4	Manual Star	Garrison	2756
132N6	ISO Procs	Jacanda	2954
132N7	Web Page	Friedman	2846
132P4	Emp. Manual	Jones	3102
132T6	Star Prot.	Garrison	2756
431Y7	New Cat.	Jones	3102
313M6	Star Price	Vance	3022
323H6	System Ord	Jacanda	2954



Project	ProjctTitle	Manager	Manager	Telf
132N4	Manual Star	Garrison	Garrison	2756
132N6	ISO Procs	Jacanda	Jacanda	2954
132N7	Web Page	Friedman	Friedman	2846
132P4	Emp. Manual	Jones	Jones	3102
132T6	Star Prot.	Garrison	Vance	3022
431Y7	New Cat.	Jones		
313M6	Star Price	Vance		
323H6	System Ord	Jacanda		



Boyce Codd Normal Form (BCNF)

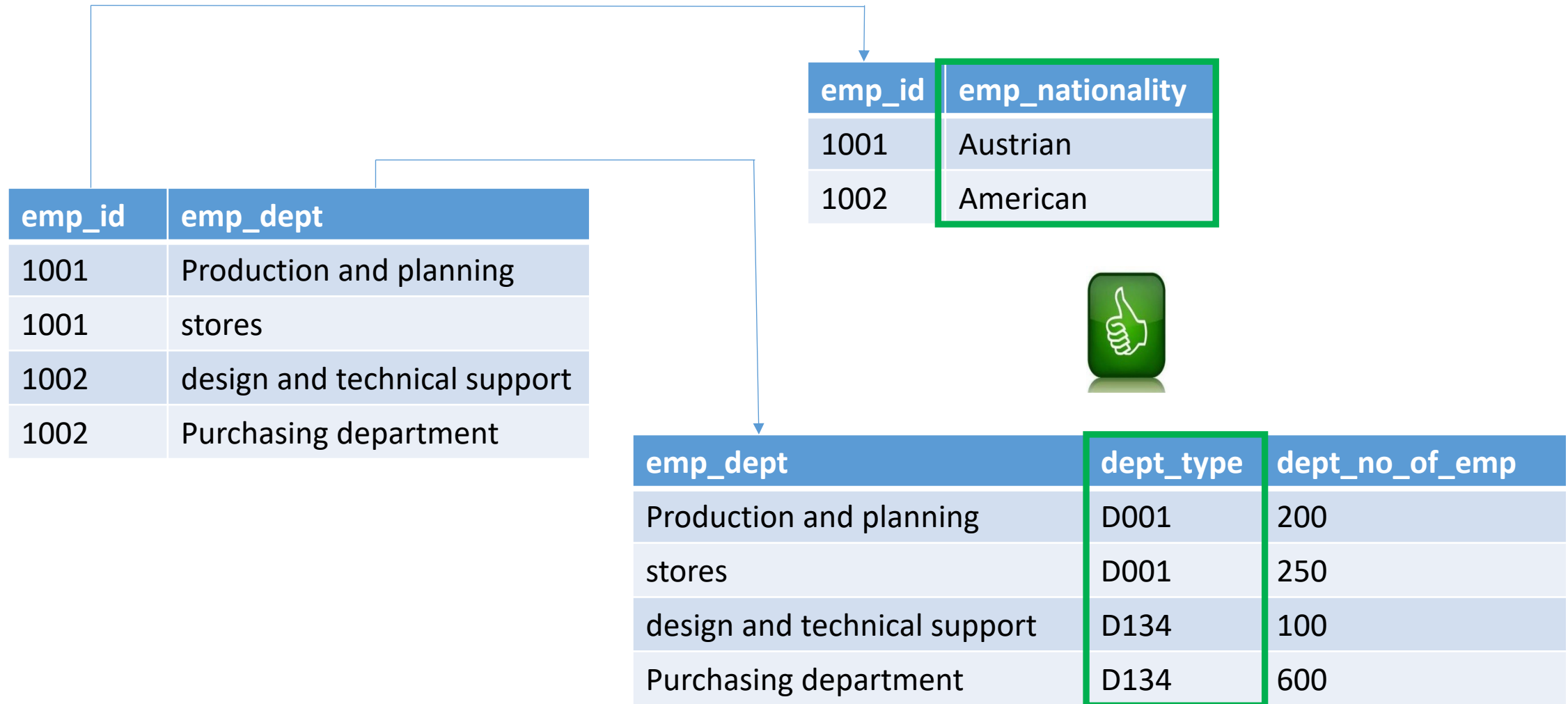
- BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency $X \rightarrow Y$, X should be the super key of the table.

emp_id	emp_country	emp_dept	dept_type	dept_no_of_emp
1001	Austrian	Production and planning	D001	200
1001	Austrian	stores	D001	250
1002	American	design and technical support	D134	100
1002	American	Purchasing department	D134	600



NORMALIZATION

Boyce Codd Normal Form (BCNF)



SESSION 9

- RELATIONAL DATABASES BASICS
- TABLES RELATIONSHIP
- NORMALIZATION
- **RELATIONAL DATABASE CASE 1**
- RELATIONAL DATABASE CASE 2
- RELATIONAL DATABASES CONSTRAINTS

RELATIONAL DATABASES CASE 1

- An International Brand wants to build a Relational Model for its **sales**.
- The Brand owns several **shops** which are the only way to sell by the moment.
- They want to register all the **employees** and their hiring dates.
- They want to register all the **sales** done in each **shop** and also they want record the **seller** of each sale.
- They also want to register all the offered **products** and the **suppliers** that supply each product.

RELATIONAL DATABASES CASE 1

- Structure the data into entities

Shop	Supplier	Product	Employee	Sale
ShopName	SupplierName	ProductName	EmployeeName	ProductName
Address	DocumentID	ProductType	EmployeeLastName	EmployeeName
Telephone	Telephone	Unit	HiringDate	EmployeeLastName
Homepage	Address	Price	DocumentID	ShopName
OpeningDate		SupplierName		Date
				Quantity

RELATIONAL DATABASES CASE 1

- **Primary Key:** For each Table, establish the column(s) that will identify one record

Shop	Supplier	Product	Employee	Sale
ShopID	SupplierID	ProductID	EmployeeID	SaleID
ShopName	SupplierName	ProductName	EmployeeName	ProductName
Address	DocumentID	ProductType	EmployeeLastName	EmployeeName
Telephone	Telephone	Unit	HiringDate	EmployeeLastName
Homepage	Address	Price	DocumentID	ShopName
OpenningDate		SupplierName		Date
				Quantity

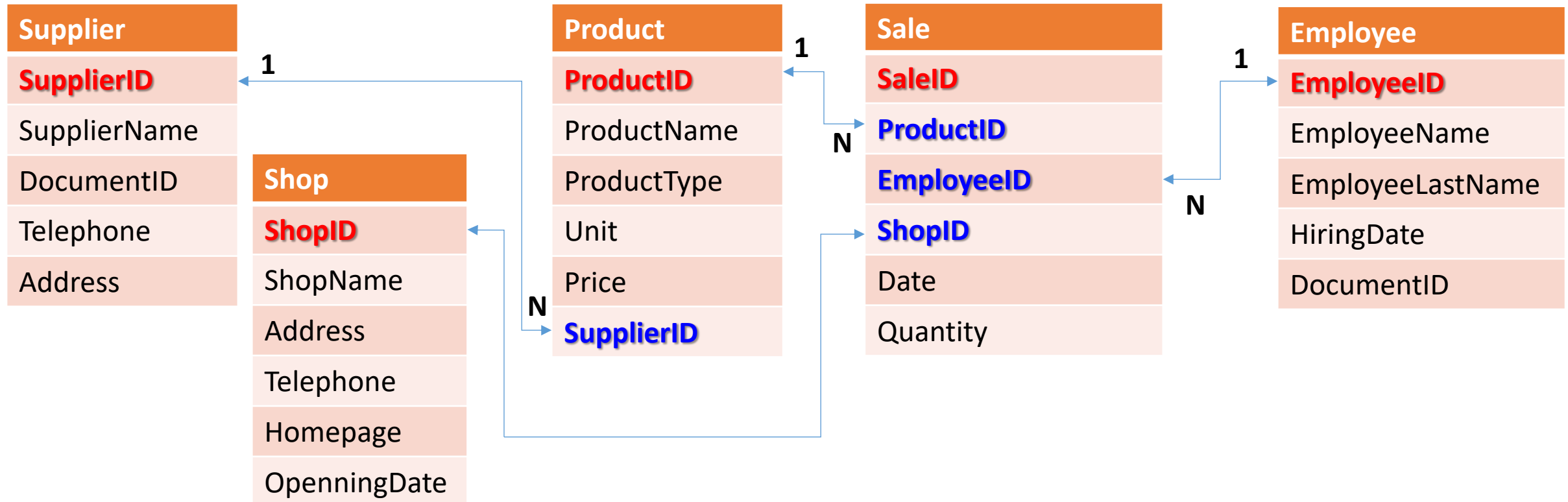
RELATIONAL DATABASES CASE 1

- **Foreign Key:** For each Table, establish the column(s) that can be reference for a Primary key from another tables

Shop	Supplier	Product	Employee	Sale
ShopID	SupplierID	ProductID	EmployeeID	SaleID
ShopName	SupplierName	ProductName	EmployeeName	ProductID
Address	DocumentID	ProductType	EmployeeLastName	EmployeeID
Telephone	Telephone	Unit	HiringDate	ShopID
Homepage	Address	Price	DocumentID	Date
OpenningDate		SupplierID		Quantity

RELATIONAL DATABASES CASE 1

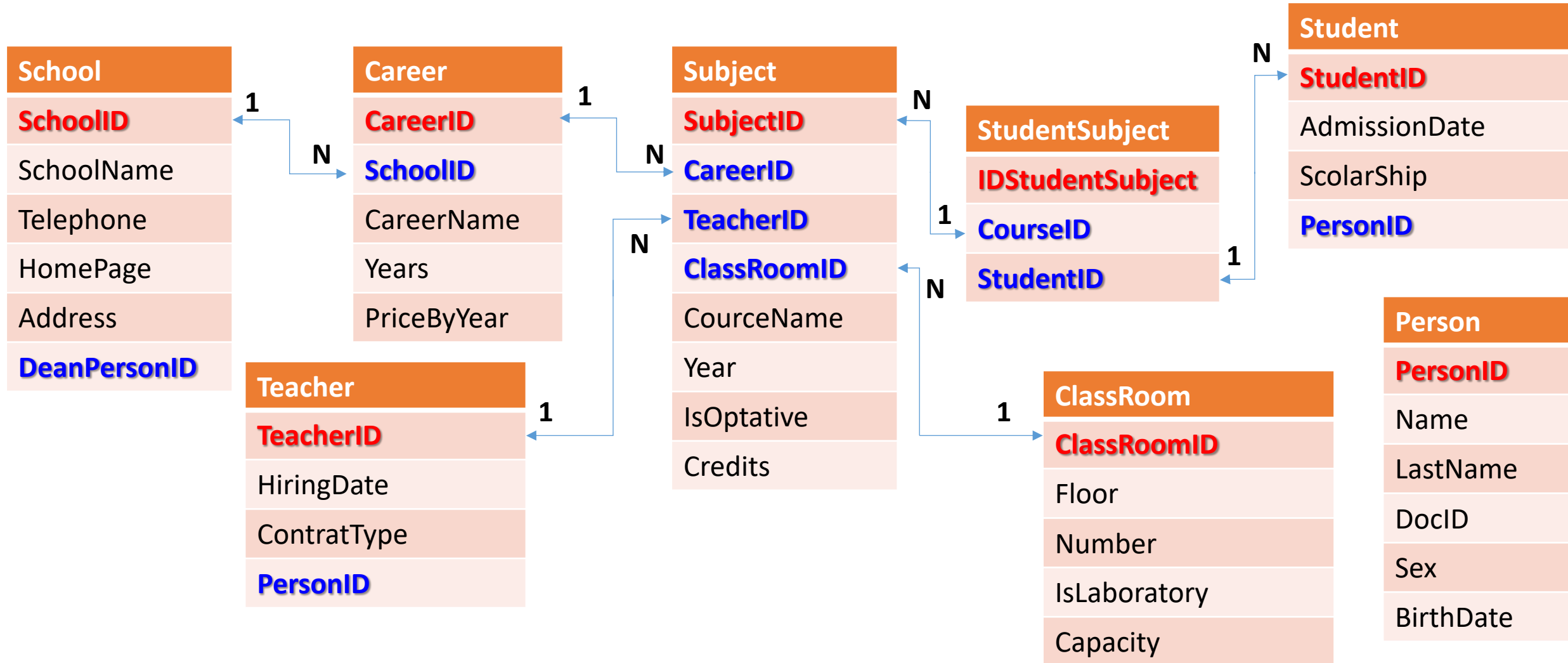
- Establish the relations. (1-N)



SESSION 9

- RELATIONAL DATABASES BASICS
- TABLES RELATIONSHIP
- NORMALIZATION
- RELATIONAL DATABASE CASE 1
- **RELATIONAL DATABASE CASE 2**
- RELATIONAL DATABASES CONSTRAINTS

RELATIONAL DATA BASES CASE 2



SESSION 9

- RELATIONAL DATABASES BASICS
- TABLES RELATIONSHIP
- NORMALIZATION
- RELATIONAL DATABASE CASE 1
- RELATIONAL DATABASE CASE 2
- **RELATIONAL DATABASES CONSTRAINTS**

ALTER TABLE – PRIMARY KEY Syntax

MySQL/Oracle

```
ALTER TABLE table ADD CONSTRAINT ConstraintName PRIMARY KEY  
(PKField1, PKField2, ...)
```

MySQL/Oracle

```
ALTER TABLE table ADD PRIMARY KEY (PKField1, PKField2, ...)
```

RELATIONAL DATABASES CONSTRAINTS

ALTER TABLE – PRIMARY KEY Example

MySQL/Oracle

```
SELECT * FROM moviesmall;
```

ID	NAME	YEAR	RANK
10920	Aliens	1986	8.2
17173	Animal House	1978	7.5

MySQL/Oracle

```
INSERT INTO moviesmall VALUES (10920, 'Aliens', 1990, 7.6);  
ROLLBACK;
```

ID	NAME	YEAR	RANK
10920	Aliens	1986	8.2
10920	Aliens	1990	7.6

MySQL/Oracle

```
ALTER TABLE moviesmall ADD CONSTRAINT pk_moviesmall PRIMARY KEY (ID);
```

MySQL/Oracle

```
INSERT INTO moviesmall VALUES (10920, 'Aliens', 1990, 7.6);
```

```
INSERT INTO moviesmall VALUES (10920, 'Aliens', 1990, 7.6)  
Informe de error -  
ORA-00001: restricción única (PAVEL.PK_MOVIESMALL) violada
```

CREATE TABLE – PRIMARY KEY Syntax

Oracle

```
CREATE TABLE table (PkField1 DataType CONSTRAINT ConstraintName PRIMARY KEY,  
Field2 DataType, Field3 DataType);
```

MySQL/Oracle

```
CREATE TABLE table (PkField1 DataType PRIMARY KEY, Field2 DataType,  
Field3 DataType);
```

MySQL/Oracle

```
CREATE TABLE table (PkField1 DataType, Field2 DataType, Field3 DataType,  
PRIMARY KEY (PkField1) );
```

MySQL/Oracle

```
CREATE TABLE table (PkField1 DataType, Field2 DataType, Field3 DataType,  
CONSTRAINT ConstraintName PRIMARY KEY (PkField1) );
```

CREATE TABLE – PRIMARY KEY Example

MySQL

```
CREATE TABLE car (  
  make VARCHAR(50),  
  model VARCHAR(100),  
  category VARCHAR(100),  
  year INTEGER,  
  CONSTRAINT pk_car PRIMARY  
  KEY (make,model) );
```

Oracle

```
CREATE TABLE car (  
  make VARCHAR2(50),  
  model VARCHAR2(100),  
  category VARCHAR2(100),  
  year NUMBER,  
  CONSTRAINT pk_car PRIMARY  
  KEY (make,model) );
```

CREATE TABLE – PRIMARY KEY Example

MySQL/Oracle

```
INSERT INTO car VALUES ('Fiat','124 Spider','Convertible',2019);
```

MAKE	MODEL	CATEGORY	YEAR
Fiat	124 Spider	Convertible	2019

MySQL/Oracle

```
INSERT INTO car VALUES ('Fiat','124 Spider','Convertible',2020);
```

```
INSERT INTO car VALUES ('Fiat','124 Spider','Convertible',2020)
Informe de error -
ORA-00001: restricción única (PAVEL.PK_CAR) violada
```

ALTER TABLE – FOREIGN KEY Syntax

MySQL/Oracle

```
ALTER TABLE table
```

```
ADD FOREIGN KEY (FKField) REFERENCES RefTable(RefField)
```

MySQL/Oracle

```
ALTER TABLE table
```

```
ADD CONSTRAINT ConstraintName
```

```
FOREIGN KEY (FKField) REFERENCES refTable(RefField)
```

ALTER TABLE – FOREIGN KEY Example

MySQL

```
CREATE TABLE car (  
  make VARCHAR(50),  
  model VARCHAR(100),  
  idcategory INTEGER,  
  year INTEGER,  
  CONSTRAINT pk_car PRIMARY KEY  
    (make,model));
```

```
CREATE TABLE category (  
  id INTEGER,  
  description VARCHAR(100),  
  CONSTRAINT pk_category PRIMARY KEY (id));
```

Oracle

```
CREATE TABLE car (  
  make VARCHAR2(50),  
  model VARCHAR2(100),  
  idcategory NUMBER,  
  year NUMBER,  
  CONSTRAINT pk_car PRIMARY KEY  
    (make,model));
```

```
CREATE TABLE category (  
  id NUMBER,  
  description VARCHAR2(100),  
  CONSTRAINT pk_category PRIMARY KEY (id));
```

ALTER TABLE – FOREIGN KEY Example

MySQL/Oracle

```
INSERT INTO category VALUES (1, 'Convertible')
```

```
INSERT INTO category VALUES (2, 'Pickup')
```

```
INSERT INTO car VALUES ('Fiat', '124 Spider', 1, 2019);
```

```
INSERT INTO car VALUES ('Ram', '1500 Classic Crew Cab', 2, 2019);
```

```
INSERT INTO car VALUES ('BMW', '1 Series', 3, 2008);
```

ID	DESCRIPTION
1	Convertible
2	Pickup

Category

MAKE	MODEL	IDCATEGORY	YEAR
Fiat	124 Spider	1	2019
Ram	1500 Classic Crew Cab	2	2019
BMW	1 Series	3	2008

Car

ALTER TABLE – FOREIGN KEY Example

MySQL/Oracle

```
ALTER TABLE car ADD CONSTRAINT fk_idcategory  
FOREIGN KEY (idcategory) REFERENCES category(id) ;
```

```
ALTER TABLE car  
ADD CONSTRAINT fk_idcategory  
FOREIGN KEY (idcategory) REFERENCES category(id)  
Informe de error -  
ORA-02298: no se puede validar (PAVEL.FK_IDCATEGORY) - claves principales no encontradas  
02298. 00000 - "cannot validate (%s.%s) - parent keys not found"  
*Cause:      an alter table validating constraint failed because the table has  
              child records.  
*Action:     Obvious
```

MySQL/Oracle

```
ROLLBACK;
```

ALTER TABLE – FOREIGN KEY Example

MySQL/Oracle

```
ALTER TABLE car ADD CONSTRAINT fk_idcategory  
FOREIGN KEY (idcategory) REFERENCES category(id);
```

MySQL/Oracle

```
INSERT INTO category VALUES (1,'Convertible')  
INSERT INTO category VALUES (2,'Pickup')  
  
INSERT INTO car VALUES ('Fiat','124 Spider',1,2019);  
INSERT INTO car VALUES ('Ram','1500 Classic Crew Cab',2,2019);
```

Category

ID	DESCRIPTION
1	Convertible
2	Pickup

Car

MAKE	MODEL	IDCATEGORY	YEAR
Fiat	124 Spider	1	2019
Ram	1500 Classic Crew Cab	2	2019

ALTER TABLE – FOREIGN KEY Example

Category

ID	DESCRIPTION
1	Convertible
2	Pickup

Car

MAKE	MODEL	IDCATEGORY	YEAR
Fiat	124 Spider	1	2019
Ram	1500 Classic Crew Cab	2	2019

MySQL/Oracle

```
INSERT INTO car VALUES ('BMW','1 Series',3,2008);
```

```
INSERT INTO car VALUES ('BMW','1 Series',3,2008)
```

```
Informe de error -
```

```
ORA-02291: restricción de integridad (PAVEL.FK_IDCATEGORY) violada - clave principal no encontrada
```

CREATE TABLE – FOREIGN KEY Syntax

Oracle

```
CREATE TABLE table (Field1 DataType PRIMARY KEY, Field2 DataType CONSTRAINT  
ConstraintName REFERENCES RefTable(RefField), Field3 DataType)
```

MySQL/Oracle

```
CREATE TABLE table (Field1 DataType PRIMARY KEY, Field2 DataType, ...,  
FOREIGN KEY (Field2) REFERENCES RefTable(RefField ))
```

MySQL/Oracle

```
CREATE TABLE table (Field1 DataType, Field2 DataType, ...,  
CONSTRAINT Constraint1Name PRIMARY KEY (Field1),  
CONSTRAINT Constraint2Name FOREIGN KEY (Field2) REFERENCES RefTable(RefField ))
```

CREATE TABLE – FOREIGN KEY Example

MySQL

```
CREATE TABLE mom (id INTEGER,  
name VARCHAR(30),  
CONSTRAINT pk_mom PRIMARY KEY (id));
```

```
CREATE TABLE child (id INTEGER,  
momid INTEGER,  
name VARCHAR(30),  
CONSTRAINT pk_child PRIMARY KEY (id),  
CONSTRAINT fk_mom FOREIGN KEY (momid)  
REFERENCES mom(id) );
```

Oracle

```
CREATE TABLE mom (id NUMBER,  
name VARCHAR2(30),  
CONSTRAINT pk_mom PRIMARY KEY (id));
```

```
CREATE TABLE child (id NUMBER,  
momid NUMBER,  
name VARCHAR2(30),  
CONSTRAINT pk_child PRIMARY KEY (id),  
CONSTRAINT fk_mom FOREIGN KEY (momid)  
REFERENCES mom(id) );
```

CREATE TABLE – FOREIGN KEY Example

MySQL/Oracle

```
INSERT INTO mom VALUES (1, 'Adele');  
INSERT INTO mom VALUES (2, 'Amy');  
INSERT INTO child VALUES (1, 1, 'George');  
INSERT INTO child VALUES (2, 1, 'Bruce');  
INSERT INTO child VALUES (3, 2, 'Bob');
```

Mom

ID	NAME
1	Adele
2	Amy

Child

ID	MOMID	NAME
1	1	George
2	1	Bruce
3	2	Bob

MySQL/Oracle

```
INSERT INTO child VALUES (4, 3, 'John');
```

```
INSERT INTO child VALUES (4,3,'John')  
Informe de error -  
ORA-02291: restricción de integridad (PAVEL.FK_MOM) violada - clave principal no encontrada
```

ACTIVATE/DEACTIVATE CONSTRAINTS

MySQL – Unique KEYS

```
SET unique_checks=0;  
SET unique_checks=1;
```

MySQL – Foreign KEYS

```
SET foreign_key_checks=0;  
SET foreign_key_checks=1;
```

Oracle

```
ALTER TABLE table  
DISABLE CONSTRAINT ConstraintName;  
  
ALTER TABLE table  
ENABLE CONSTRAINT ConstraintName;
```

DROP PRIMARY KEY CONSTRAINT Syntax

MySQL/Oracle

```
ALTER TABLE table  
DROP PRIMARY KEY
```

Oracle

```
ALTER TABLE table  
DROP CONSTRAINT ConstraintName
```

DROP FOREIGN KEY CONSTRAINT Syntax

MySQL

```
ALTER TABLE table  
DROP FOREIGN KEY ConstraintName
```

Oracle

```
ALTER TABLE table  
DROP CONSTRAINT ConstraintName
```