

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 1 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

The Relational Data Model

The data structures of the relational model

- Attributes and domains
- Relation schemas and database schemas (decomposition)
- The universal relation schema assumption
- Relations and databases
- First normal form (1NF)

Supplier-Parts-DB

SNUM	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Supplier table

PNUM	PNAME	COLOUR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

Part table

JNUM	JNAME	CITY
J1	Sorter	Paris
J2	Display	Rome
J3	OCR	Athens
J4	Console	Athens
J5	RAID	London
J6	EDS	Oslo
J7	Tape	London

Project table

[Home Page](#)

[Title Page](#)

◀ ▶

◀ ▶

Page 2 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 3 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

SNUM	PNUM	JNUM	QTY
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S2	P5	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P5	J5	500
S5	P5	J7	100
S5	P5	J4	400
S5	P1	J4	100
S5	P3	J4	200
S5	P4	J4	800
S5	P6	J4	500
S5	P6	J2	200

Supply table

Attributes and Domains

Assumptions:

- U is a **universal** set of attributes
- D is a the **underlying database domain** of constant values.
- The domain of A in U , is $DOM(A)$ is a subset of D .
- **Unique Name Assumption (UNA)**: two constants c_1, c_2 in D are equal if and only if they are syntactically equal.

E.g. “Mark” = “Mark” but “Mark” \neq “John” and “Mark” \neq 143.

Relation schemas and Database Schemas

A **relation schema** (or **table header**) R has the following components:

- A **relation symbol** R , which is the name of the schema.
- A **similarity type** $\text{type}(R)$, which denotes the number of attributes of R .
- A set of **attributes** (or **column headers**) $\{att(1), \dots, att(\text{type}(R))\}$, denoted by $\text{schema}(R)$.

E.g. SUPPLIER is a relation symbol,

$\text{type}(\text{SUPPLIER}) = 4$, and

$\text{schema}(\text{SUPPLIER}) = \{att(1), att(2), att(3), att(4)\}$,

where $att(1) = \text{SNUM}$, $att(2) = \text{SNAME}$, $att(3) = \text{STATUS}$,

and $att(4) = \text{CITY}$.

A **database schema \mathbf{R}** is a collection $\{R_1, \dots, R_n\}$ of relation schemas (table headers).

E.g. The database schema of the Supplier-parts-DB is $\{\text{SUPPLIER, PART, PROJECT, SUPPLY}\}$.

Notation.

$\text{schema}(\mathbf{R})$ is the union of all $\text{schema}(R_i)$, R_i in \mathbf{R} .

$\text{schema}(\text{Supplier-Parts-DB}) = \{\text{SNUM, SNAME, STATUS, CITY, PNUM, PNAME, COLOUR, WEIGHT, JNUM, JNAME, QTY}\}$

- A relation schema R is in **First Normal Form** (1NF) if all the domains of attributes A_i in $\text{schema}(R)$ are atomic. (I.e. non-decomposable by the DBMS.)
- A database schema \mathbf{R} is in 1NF if all the relation schemas R_i in \mathbf{R} are in 1NF.

A non-1NF version of the **SUPPLY** table

SNUM	PNUM	JNUM	QTY
S1	P1	J1	200
		J4	700
S2	P3	J1	400
		J2	200
		J3	200
		J4	500
		J5	600
		J6	400
		J7	800
S2	P5	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300
		J7	300
S5	P2	J2	200
		J4	100
S5	P5	J5	500
		J7	100
		J4	400
S5	P1	J4	100
S5	P3	J4	200
S5	P4	J4	800
S5	P6	J2	200
		J4	500

Supply-Non-1NF table

[Home Page](#)

[Title Page](#)



Page 7 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

From now on we will assume that database schemas are in 1NF.

The justification for this assumption is:

1. The semantics of 1NF are easy to understand (e.g. ADDRESS vs. ST_NO, ST_NAME and CITY).
2. 1NF makes it easier to formalise the relational model; flat relations (**SUPPLY table**) vs. nested relations (**SUPPLY-NON-1NF table**).

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 9 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

The Universal Relation Schema Assumption (URSA)

If an attribute A appears both in $\text{schema}(R_i)$ and in $\text{schema}(R_j)$ then it has the same meaning.

E.g. use SNAME to mean supplier name,
use PNAME to mean part name and
use JNAME to mean project name.

Question. Does CITY violate the URSA ?

Relations and Databases

A **tuple** (or **row**) over R , with $\text{schema}(R) = \{A_1, \dots, A_m\}$ is a member of

$$\text{DOM}(A_1) \times \dots \times \text{DOM}(A_m),$$

where \times is the Cartesian product operator.

A **relation** (or **table**) over R is a finite set of tuples over R .

A **database** d over \mathbf{R} is a collection $\{r_1, \dots, r_n\}$ of relations r_i over R_i .

Cartesian Product Example

Boy		Girl
Joseph		Tamara
Nimrod		Lara
Sasha		Maria

Boy and Girl Tables

Boy	Girl
Joseph	Tamara
Joseph	Lara
Joseph	Maria
Nimrod	Tamara
Nimrod	Lara
Nimrod	Maria
Sasha	Tamara
Sasha	Lara
Sasha	Maria

Pairs Table = Boy \times Girl

[Home Page](#)

[Title Page](#)

◀◀ ▶▶

◀ ▶

Page 11 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 12 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

First Normal Form (1NF)

Relations over 1NF relation schemas are called **1NF relations** (or **flat relations** or simply relations).

1NF relations are advantageous since they have

1. A simple tabular representation.
2. Simple query languages.
3. A simple set of fundamental integrity constraints.

[Home Page](#)

[Title Page](#)

◀ ▶

◀ ▶

Page 13 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Synonymous notation:

- relation schema = *table header*
- database schema = *database headers*
- relation = *table*
- database = *database tables*
- attribute = *column header*
- attribute value = *table cell*
- tuple = *row*

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 14 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Summary of the properties of tables:

- Tables names in a database are distinct.
- Column names in a table are distinct.
- The order of columns and rows in a table is *not* important.
- No two rows in a table are the same, i.e. a table does *not* contain duplicate rows.
- Table cells are atomic.

Projection

Let R be a relation schema, X be a subset of schema(R) be a set of attributes and t be a tuple over R .

The **projection** of t onto X , denoted by $t[X]$, is the collection of attribute values of t under the column headers in X , i.e. the restriction of t to X .

E.g. if

$$t = (S1, \text{Smith}, 20, \text{London})$$

then

$$t[\text{City}] = (\text{London})$$

and

$$t[\text{SNUM}, \text{STATUS}] = (S1, 20).$$

Home Page

Title Page

◀ ▶

◀ ▶

Page 16 of 27

Go Back

Full Screen

Close

Quit

Null Values

We must allow for *missing* or *incomplete* information by allowing *null* values as place holders for non-null constants.

E.g. An Employee's address is *unknown*.

E.g. An Employees's spouse *does not exist*.

The distinguished place holder *null* will be used as a null value.

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 17 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Superkey

Definition of a superkey for R .

A subset SK of $\text{schema}(R)$ is a *superkey* for R if
for all relation instances r of R , the projection $t[SK]$ of any tuple t over R uniquely identifies a single tuple in r .

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 18 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Key

Definition of a key for R . A (candidate) *key* for a relation schema R is a superkey for R having a *minimal number of column headers*.

Definition of primary key of R . A *primary key* for R is one of the candidate keys, which is designated by the database designer as being primary.

Question. What are the candidate keys for the **Supplier-Parts DB** tables ?

Foreign Key

Let \mathbf{R} be a database schema, R_1, R_2 in \mathbf{R} and assume that K is the primary key of R_2 .

Definition of a foreign key.

FK a subset of R_1 is a *foreign key* for R_1 referencing the primary key K of R_2 if the following condition holds:

for all database instances $d = \{r_1, r_2, \dots, r_n\}$ of \mathbf{R} and for all tuples t_1 in r_1 , if $t_1[FK]$ does not contain any null values, then there exist a tuple t_2 in r_2 such that $t_1[FK] = t_2[K]$.

Question. What are the foreign keys for the **Supplier-Parts DB** tables ?

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 20 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

The First Fundamental Integrity Constraint of the Relational Model

Let K be the primary key of R_1 .

Definition of entity integrity. Primary key values $t[K]$ of tuples t in relations over R_1 should **not** contain null values.

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 21 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

The Second Fundamental Integrity Constraint of the Relational Model

Let FK be a foreign key for R_2 referencing K .

Definition of referential integrity. If all of the foreign key values $t[FK]$ of a tuple t in a relation over R_2 are all *non-null*, then $t[FK]$ are primary key values for K in the referenced relation over R_1 .

Checking for Entity Integrity

The following algorithm checks whether *entity integrity* is satisfied in a relation r over R with primary key X subset of $\text{schema}(R)$.

Algorithm 1 (Check_Primary_Key(r , X))

```
1. begin
2.   for all tuples  $t$  in  $r$  do
3.     if there exists  $A$  in  $X$  such that  $t[A]$  is null then
4.       return NO;
5.     end if
6.     for all tuples  $u$  in  $r$  minus  $\{t\}$  do
7.       if  $u[X] = t[X]$  then
8.         return NO;
9.       end if
10.    end for
11.  end for
12.  return YES;
13. end.
```

Checking for Referential Integrity

The following algorithm checks whether *referential integrity* is satisfied in a database $d = \{r, s\}$ over $\{R, S\}$, with foreign key X subset of $\text{schema}(R)$ matching primary key Y subset of $\text{schema}(S)$.

Algorithm 2 (Check Foreign Key(d, X, Y))

1. **begin**
2. **for all** tuples t in r **do**
3. **if** for all A in X , $t[A]$ is not *null* **then**
4. **if** there does not exist u in s such that $u[Y] = t[X]$ **then**
5. **return** NO;
6. **end if**
7. **end if**
8. **end for**
9. **return** YES;
10. **end.**

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 24 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Modeling Entities in the Relational Model

An entity type E , having attributes, A_1, \dots, A_m is modeled by a relation schema R , with $\text{schema}(R) = \{A_1, \dots, A_m\}$.

★ The primary key of R is the primary key of E .

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 25 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Modeling Relationships in the Relational Model

A many-to-many relationship type M having primary key K_1K_2 is modeled by a relation schema R , with $\text{schema}(R) = K_1K_2$.

★ The primary key of R is K_1K_2 .

[Home Page](#)

[Title Page](#)



Page 26 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Question. What happens if a relationship type is many-to-one or one-to-one ?

[Home Page](#)[Title Page](#)[Page 27 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Answer. No new relation schemas need to be defined but foreign keys need to be present in the appropriate relation schemas.

★ For a many-to-one relationship type from E_2 to E_1 , where K_1 is the primary key of E_1 , K_1 must be a subset of $\text{schema}(R_2)$, where R_2 models E_2 .

$\Rightarrow K_1$ is a foreign key for R_2 referencing K_1 in R_1 , where R_1 models E_1 .