DOKUZ EYLÜL UNIVERSITY
DEPARTMENT OF STATISTICS

BIL3203 – DATABASE MANAGEMENT

Alper VAHAPLAR

2020 – 2021 ©

What is "*Database*"?

- is a *base* that *data* are kept.
- A database is a collection of information that is organized so that it can easily be accessed, managed, and updated.
- A collection of regular and related data cleared from mistakes and redundancies, and stored to serve some applications.

Databases – History

- 1960's
  - Data Collections, Hierarchical DBMS, network DBMS
- 1970's
  - Relational Data Models
- 1980's
  - RDBMS, Object Oriented Databases,
- 1990 – 2000's
  - XML Databases, Data Warehouses, Multimedia Databases, Data Mining

Data Models

- Data modeling is a method used to define and analyze data requirements needed to support the business processes of an organization and by defining the data structures and the relationships between data elements.

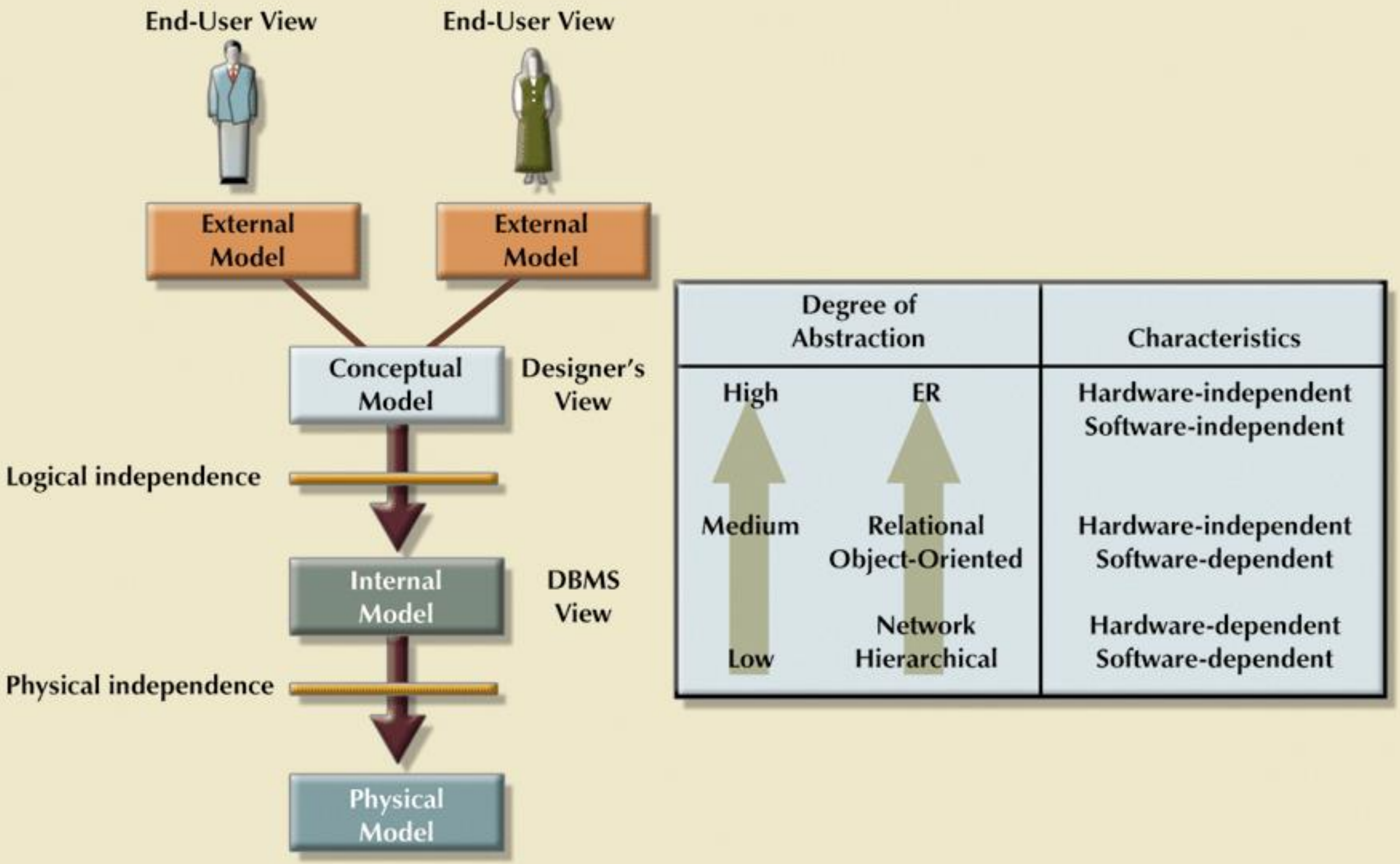
- A data model is a collection of descriptions of data structures and their contained fields, together with the operations or functions that manipulate them.

Data Models

- A data model,
  - is used to express the data in an understandable manner.
  - is a collection of concepts used to describe the structure of a database.
  - simple visualization of complex real world data.
  - helps the interaction among database designer, programmer and the end user.
  - provides different views for end users.
  - organizes the data for various types of users.

Data Models

FIGURE 2.9 Data abstraction levels



Degree of Abstraction		Characteristics
High	ER	Hardware-independent Software-independent
Medium	Relational Object-Oriented	Hardware-independent Software-dependent
Low	Network Hierarchical	Hardware-dependent Software-dependent

Data Models

- External Model
  - End user view to the data environment.
- Conceptual Model
  - Represents the global view of whole database.
- Internal Model
  - View of database to the DBMS side
- Physical Model
  - Dependence on hardware and software.

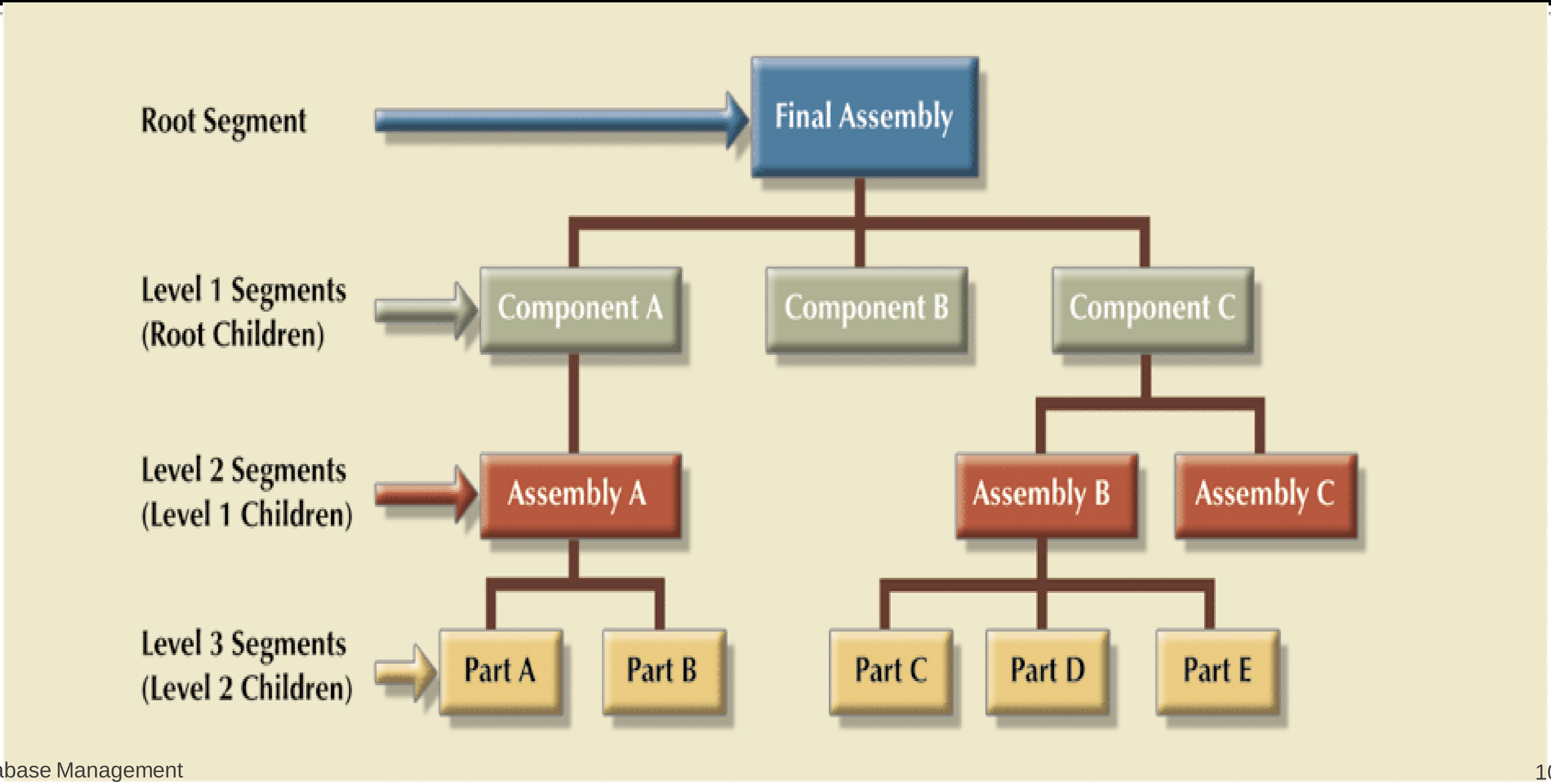
Data Models – History

- Hierarchical
- Network
- Relational
- Entity Relationship
- Object Oriented

Hierarchical Data Models

- Arised to handle production projects with huge volume in data.
- “Tree” shape,
- Constructed of “segments” or “levels”,
- Contains 1:M associations.

Hierarchical Data Models



Hierarchical Data Models

- Advantages
  - Pioneered to advanced data models,
  - Provoked database approach,
  - Many applications have been developed

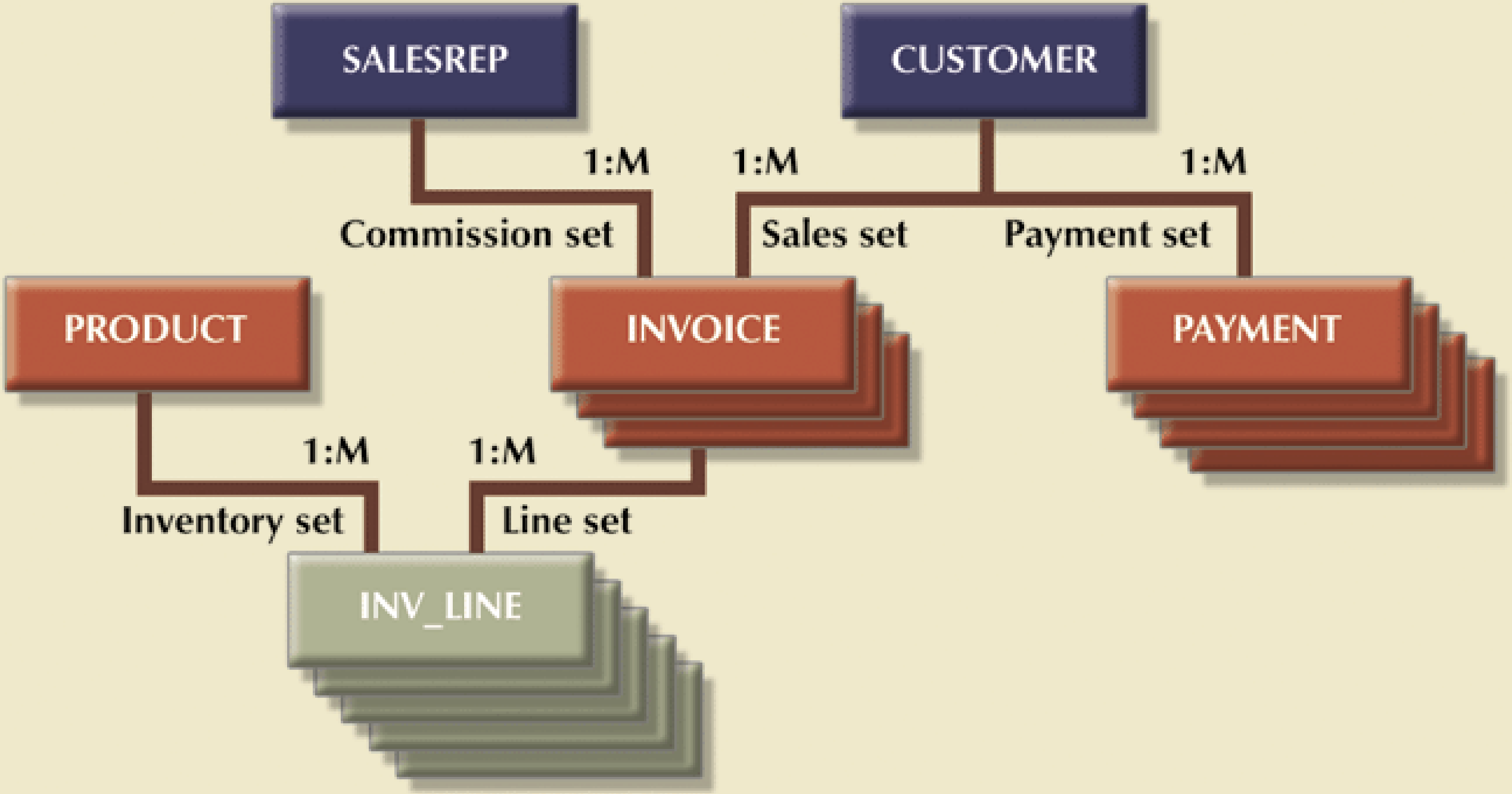
- Disadvantages
  - Hard and complex to manage and implement,
  - Limited application opportunity
  - Lack of standards,
  - No structural independency

Network Data Models

- Aims to define complex data relationships effectively,
- Works to improve the performance of database,
- Look similar to hierarchical model in shape,
- Collection of in 1:M associated records,
- Contains two types of records:
  - Owner (parent in hierarchical)
  - Member (child in hierarchical)

Network Data Models

FIGURE 2.2 A network data model



Network Data Models

- Disadvantages
  - Generally cumbersome,
  - Programmer is overloaded for queries,
  - A structural change in databases couses big damages in programs,
  - Big delays in responses.

Relational Data Model

- The Relational Model,
- 1969 – Edgar Frank Codd (IBM),
- “Very ingenious, but impractical” – 1970 ,
- Conceptually simple and plain,
- Microcomputers can successfully implement,
- Hides the complexity of the model from the user.

Elements of Relational Data Models

- Entity
  - An entity is any individual object or event in the system that we want to model and store information about .

- Attribute
  - A recordable property of an entity,

- Key
  - A key is a data item/property that uniquely identifies individual occurrences or an entity type.

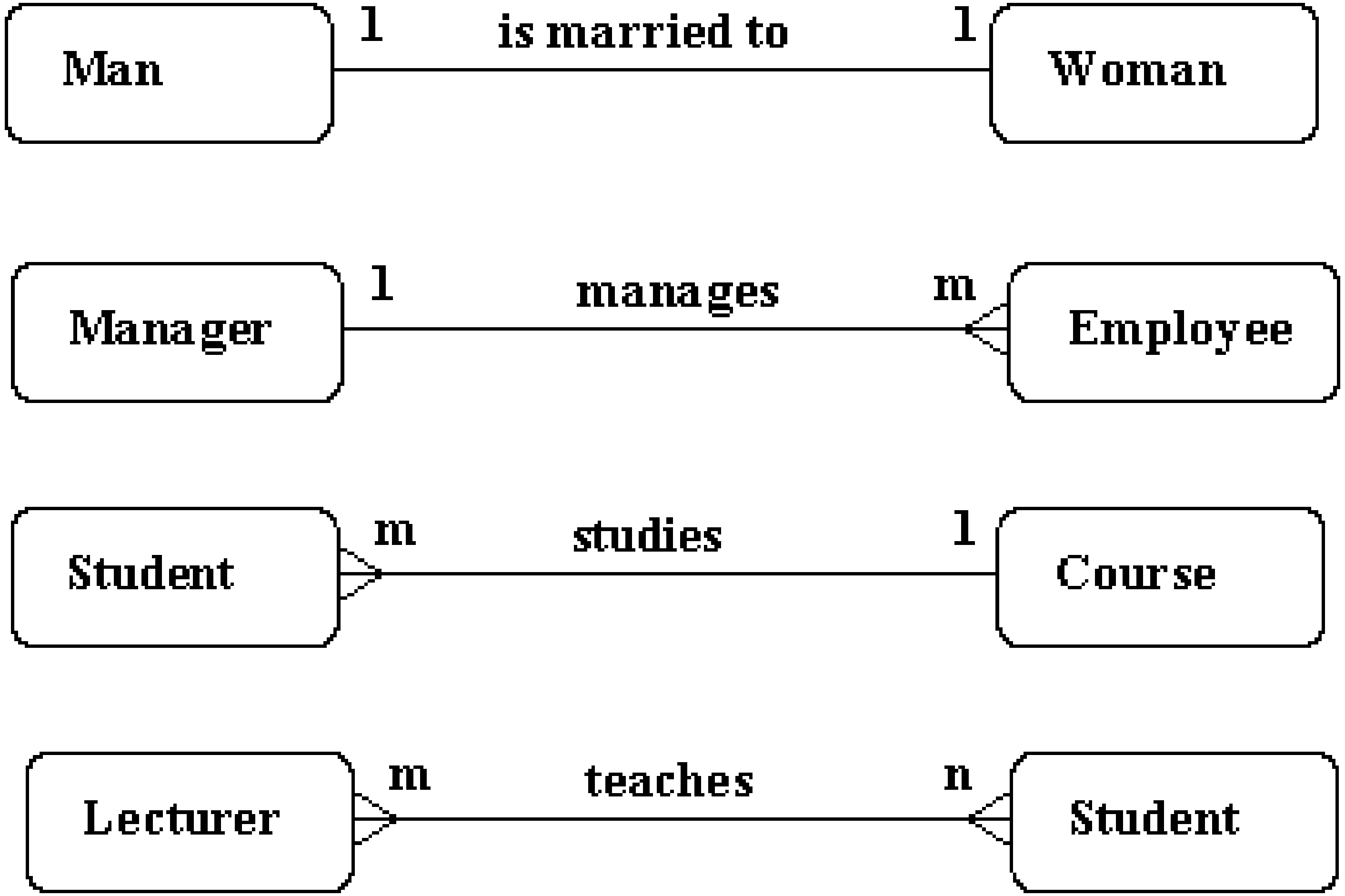
Elements of Relational Data Models

- Examples :
  - Entity
    - Student
  - Attribute
    - Student number, name, surname, birthdate, address, ...
  - Key
    - Student Number

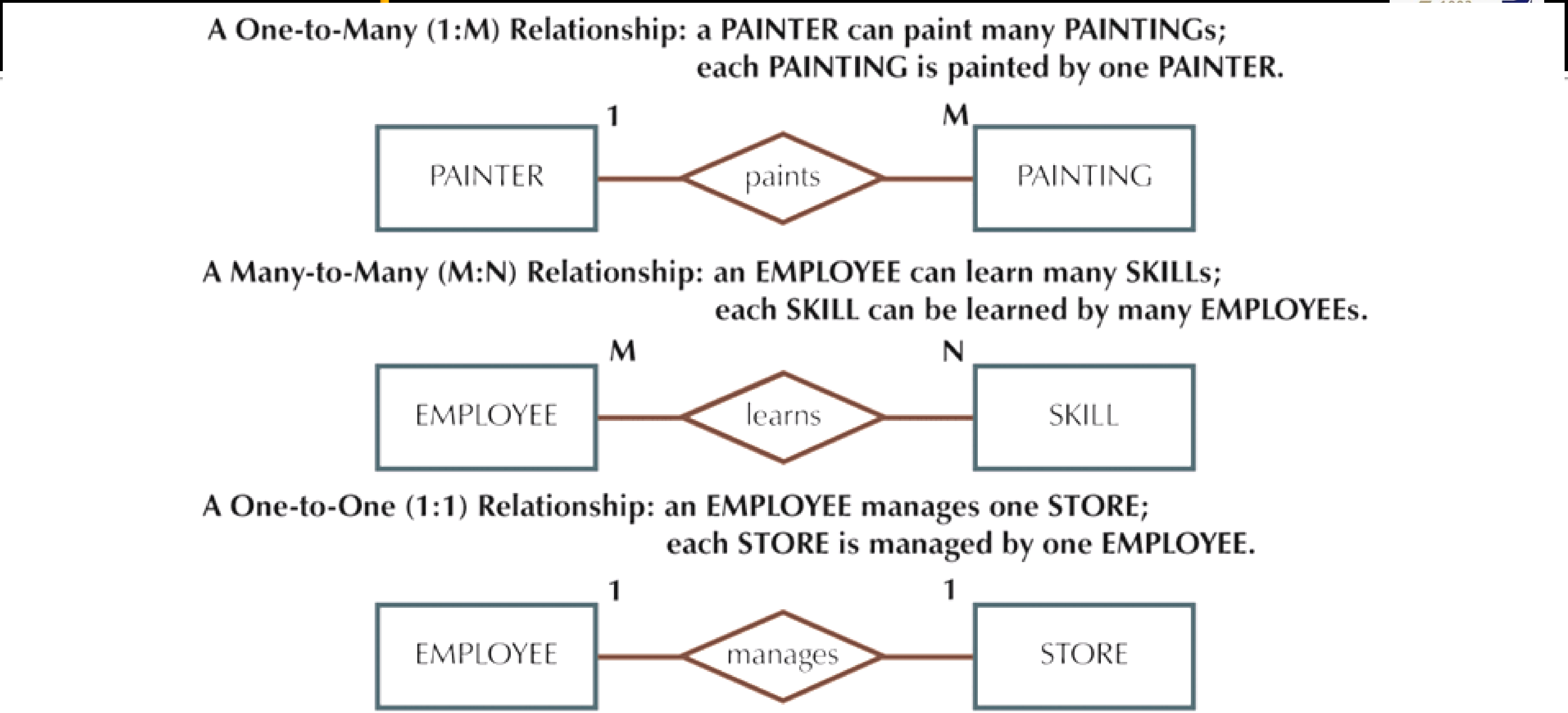
Elements of Relational Data Models

- Relationships:
  - A *relationship* is an association of two or more entities where the association includes one entity from each participating entity type.
- Types of relationships:
  - One – to – one → (1:1)
  - One – to – many→ (1:M)
  - Many – to – many → (M:N)

Relationships – Examples



Relationships



A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

Homework

- Find entity couples in relationships of
  - 1:1,
  - 1:M,
  - M:N

  association types,
- Define the attributes of these entities,
- Decide which is the key attribute.

Definitions…

- Tuple :
  - Values constructing a row, (record)
- Degree :
  - Number of attributes in a tuple
- Cardinality :
  - Number of tuples in a relation
- Domain :
  - Set of possible values for an attribute

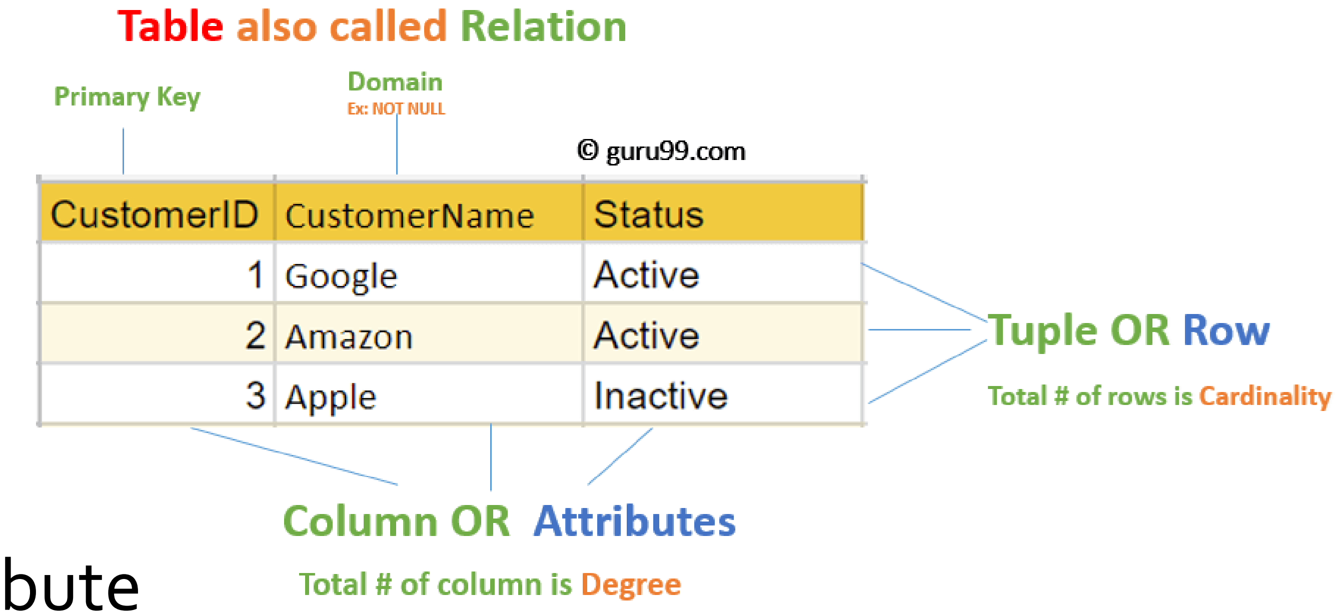


Table also called Relation

Primary Key

Domain
Ex: NOT NULL

© guru99.com

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

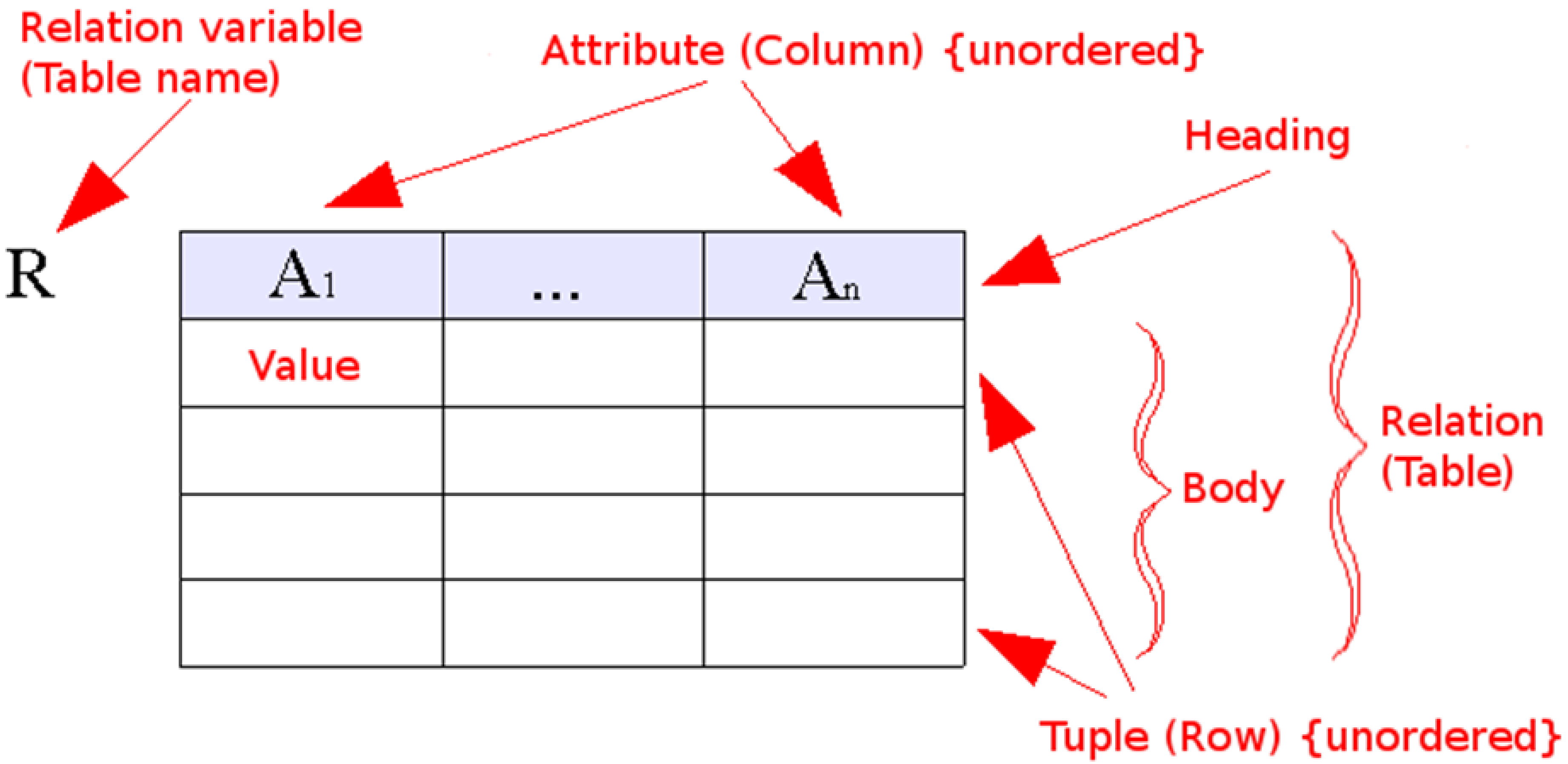
Tuple OR Row

Total # of rows is Cardinality

Column OR Attributes

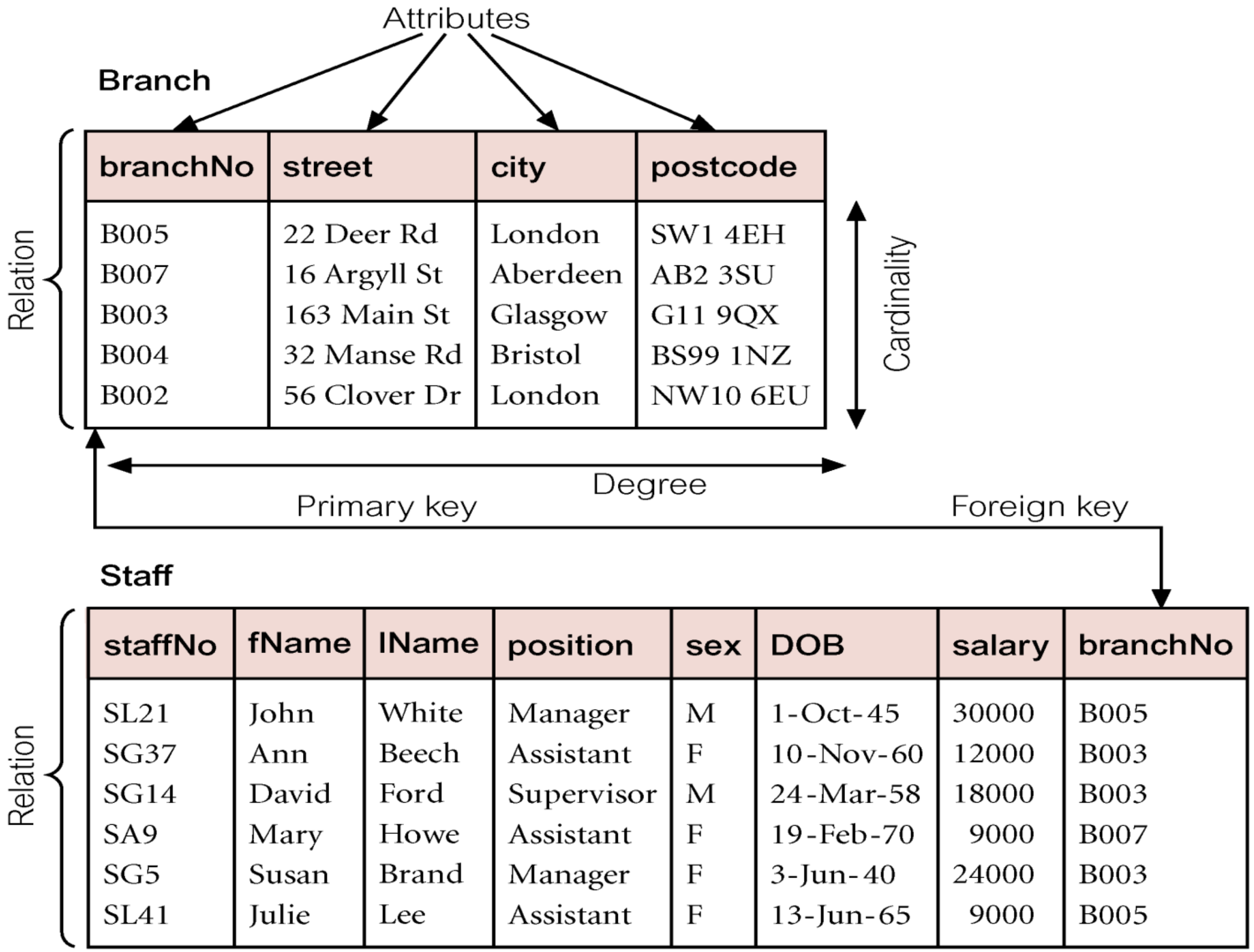
Total # of column is Degree

Relational Data Model



A_1	…	A_n
Value		

Definitions...



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Definitions…

Attribute

Öğrenciler

No	AdSoyad	Sınıf	DogYili	DogYeri
123	Ali Veli	3	1982	Muğla
456	Hasan Hüseyin	3	1981	Afyon
789	Ayşe Fatma	2	1983	İzmir
102	Ahmet Mehmet	3	1981	Aydın
260	Fuat Murat	2	1983	Manisa
346	Lale Jale	3	1982	İzmir

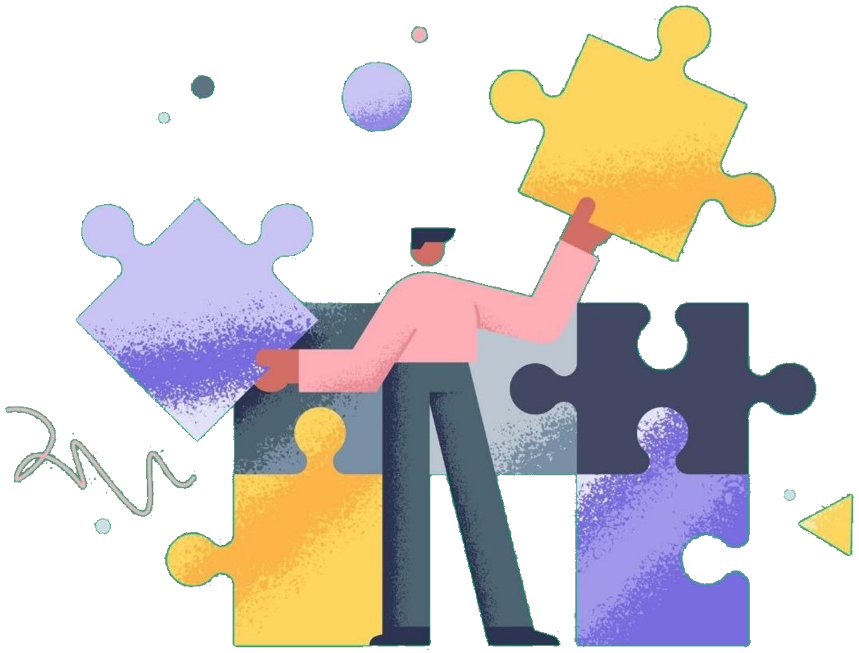
Tuple

Normalization

- Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

Normalization

- Process of assigning attributes to entities.
- Aims:
  - reducing data redundancy,
  - eliminating anomalies,
  - producing controlled redundancy to link tables.



Normalization

- There are a few rules for database normalization. Each rule is called a "**normal form**." If the first rule is observed, the database is said to be in "first normal form." If the first three rules are observed, the database is considered to be in "third normal form." Although other levels of normalization are possible, third normal form is considered the highest level necessary for most applications.

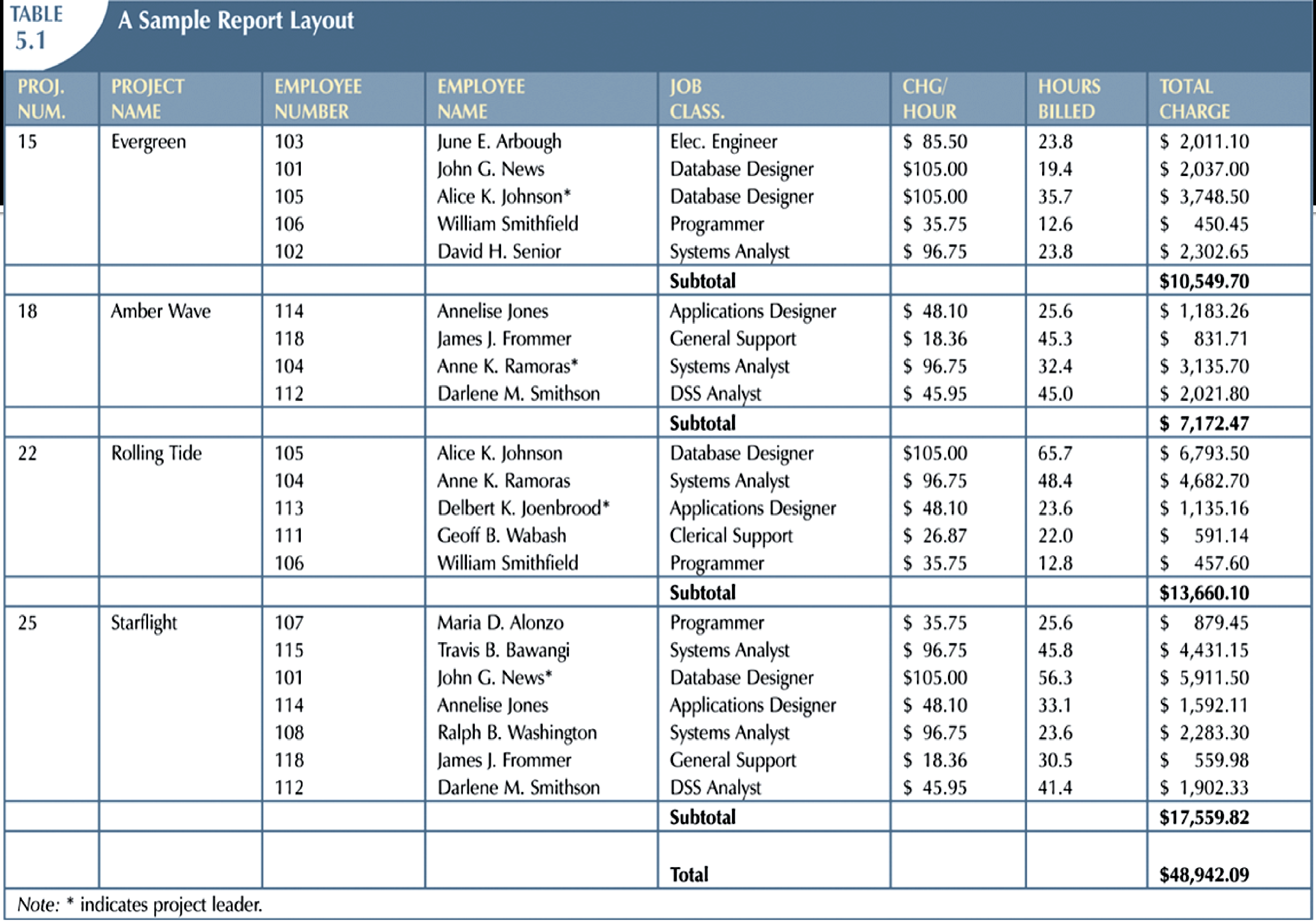


TABLE 5.1 A Sample Report Layout

PROJ. NUM.	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS.	CHG/ HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$ 85.50	23.8	$ 2,011.10
		101	John G. News	Database Designer	$105.00	19.4	$ 2,037.00
		105	Alice K. Johnson*	Database Designer	$105.00	35.7	$ 3,748.50
		106	William Smithfield	Programmer	$ 35.75	12.6	$ 450.45
		102	David H. Senior	Systems Analyst	$ 96.75	23.8	$ 2,302.65
				Subtotal			**$10,549.70**
18	Amber Wave	114	Annelise Jones	Applications Designer	$ 48.10	25.6	$ 1,183.26
		118	James J. Frommer	General Support	$ 18.36	45.3	$ 831.71
		104	Anne K. Ramoras*	Systems Analyst	$ 96.75	32.4	$ 3,135.70
		112	Darlene M. Smithson	DSS Analyst	$ 45.95	45.0	$ 2,021.80
				Subtotal			**$ 7,172.47**
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	65.7	$ 6,793.50
		104	Anne K. Ramoras	Systems Analyst	$ 96.75	48.4	$ 4,682.70
		113	Delbert K. Joenbrood*	Applications Designer	$ 48.10	23.6	$ 1,135.16
		111	Geoff B. Wabash	Clerical Support	$ 26.87	22.0	$ 591.14
		106	William Smithfield	Programmer	$ 35.75	12.8	$ 457.60
				Subtotal			**$13,660.10**
25	Starflight	107	Maria D. Alonzo	Programmer	$ 35.75	25.6	$ 879.45
		115	Travis B. Bawangi	Systems Analyst	$ 96.75	45.8	$ 4,431.15
		101	John G. News*	Database Designer	$105.00	56.3	$ 5,911.50
		114	Annelise Jones	Applications Designer	$ 48.10	33.1	$ 1,592.11
		108	Ralph B. Washington	Systems Analyst	$ 96.75	23.6	$ 2,283.30
		118	James J. Frommer	General Support	$ 18.36	30.5	$ 559.98
		112	Darlene M. Smithson	DSS Analyst	$ 45.95	41.4	$ 1,902.33
				Subtotal			**$17,559.82**
				Total			**$48,942.09**

Note: * indicates project leader.

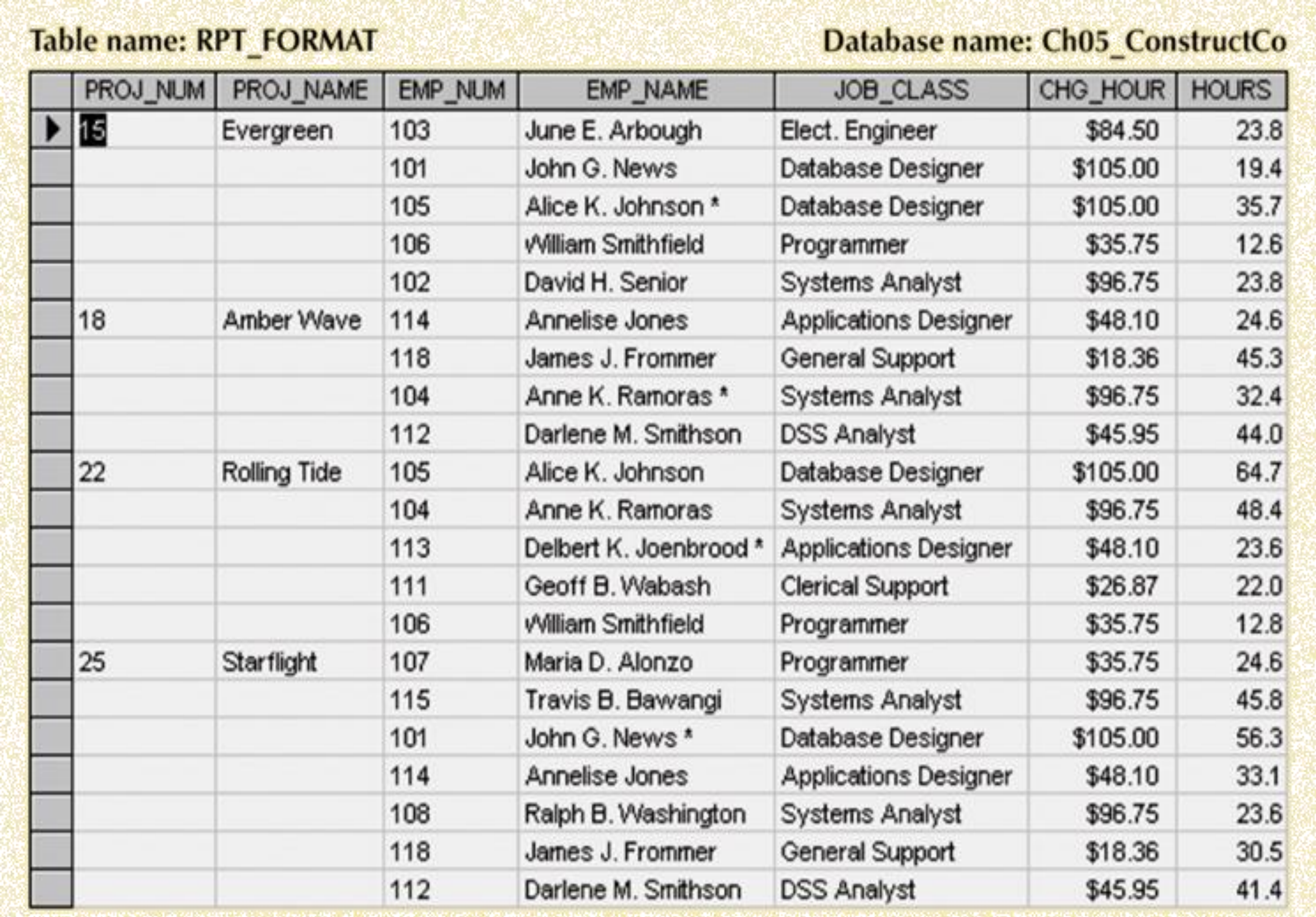


Table name: RPT_FORMAT

Database name: Ch05_ConstructCo

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
		118	James J. Frommer	General Support	$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0
		106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
		101	John G. News *	Database Designer	$105.00	56.3
		114	Annelise Jones	Applications Designer	$48.10	33.1
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6
		118	James J. Frommer	General Support	$18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

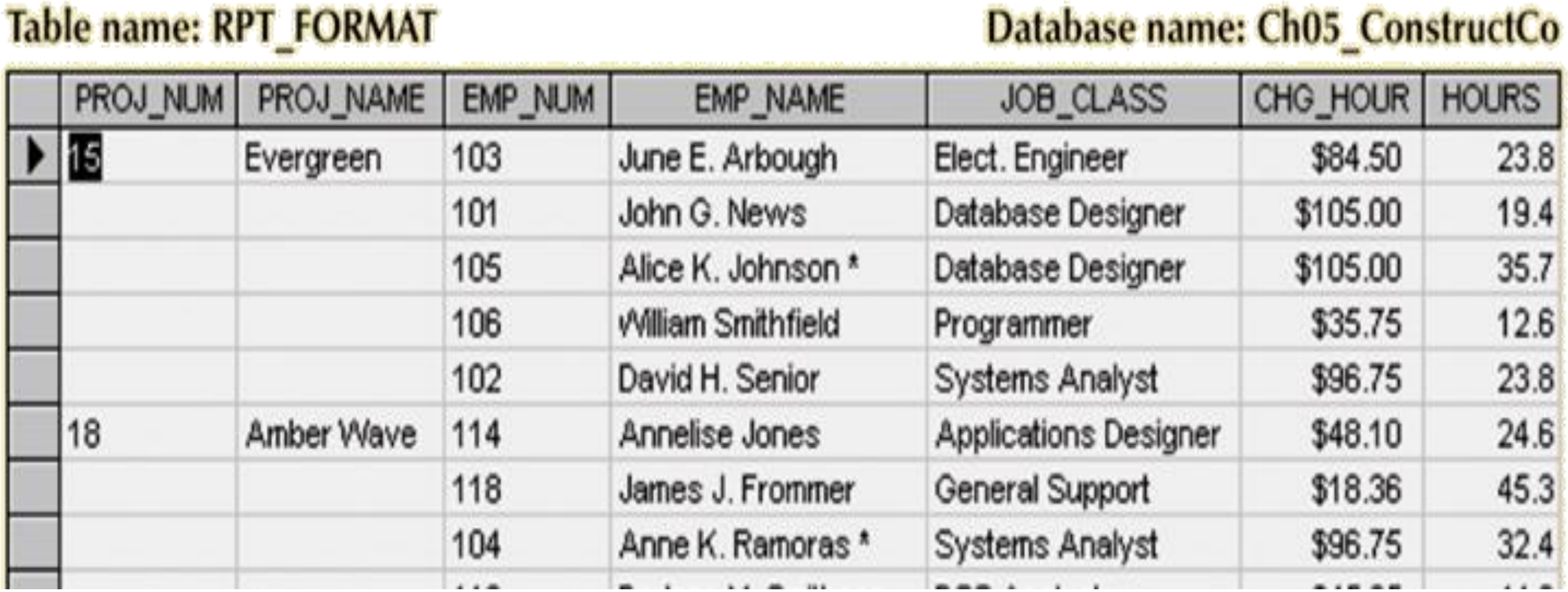


Table name: RPT_FORMAT

Database name: Ch05_ConstructCo

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
		118	James J. Frommer	General Support	$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4

RPT_FORMAT (PROJ_NUM, PROJ_NAME, EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR, HOURS)

Normalization

- Rules:
  - Each relation (table) must represent one and only one entity.
  - No data must repeat redundantly in different relations.
  - All the attributes must be dependant on the key attribute.

Normalization

- **First Normal Form**
  - Eliminate repeating groups in individual tables.
  - Create a separate table for each set of related data.
  - Identify each set of related data with a primary key.
- **Second Normal Form**
  - Create separate tables for sets of values that apply to multiple records.
  - Relate these tables with a foreign key.
- **Third Normal Form**
  - Eliminate fields that do not depend on the key.

Normalization

- **First Normal Form**
  - Eliminate repeating groups in individual tables.
  - Create a separate table for each set of related data.
  - Identify each set of related data with a primary key.

Normalization

- First Normal Form
  - **RPT_FORMAT** (Proj_num, Proj_name,{Emp_num, Emp_name, Job_class, Chg_hour, Hours})
    - PROJ (Proj_num, Proj_name)
    - EMPLOYEE (Emp_num, Emp_name, Job_class, Chg_hour, Hours)

Normalization

- **Second Normal Form**
  - Remove Partial Dependencies
  - Create separate tables for sets of values that apply to multiple records.
  - Relate these tables with a foreign key.
  - Each attribute must be FULLY DEPENDANT to the key attribute.

Normalization

- **Second Normal Form**
  - PROJ (Proj_num, Proj_name)
  - EMPLOYEE (Emp_num, Emp_name, Job_class, Chg_hour, Hours)
    - EMPLOYEE (Emp_num, Emp_name, Job_class, Chg_hour)
    - ASSIGN (Emp_num, Proj_num, Hours)

Normalization

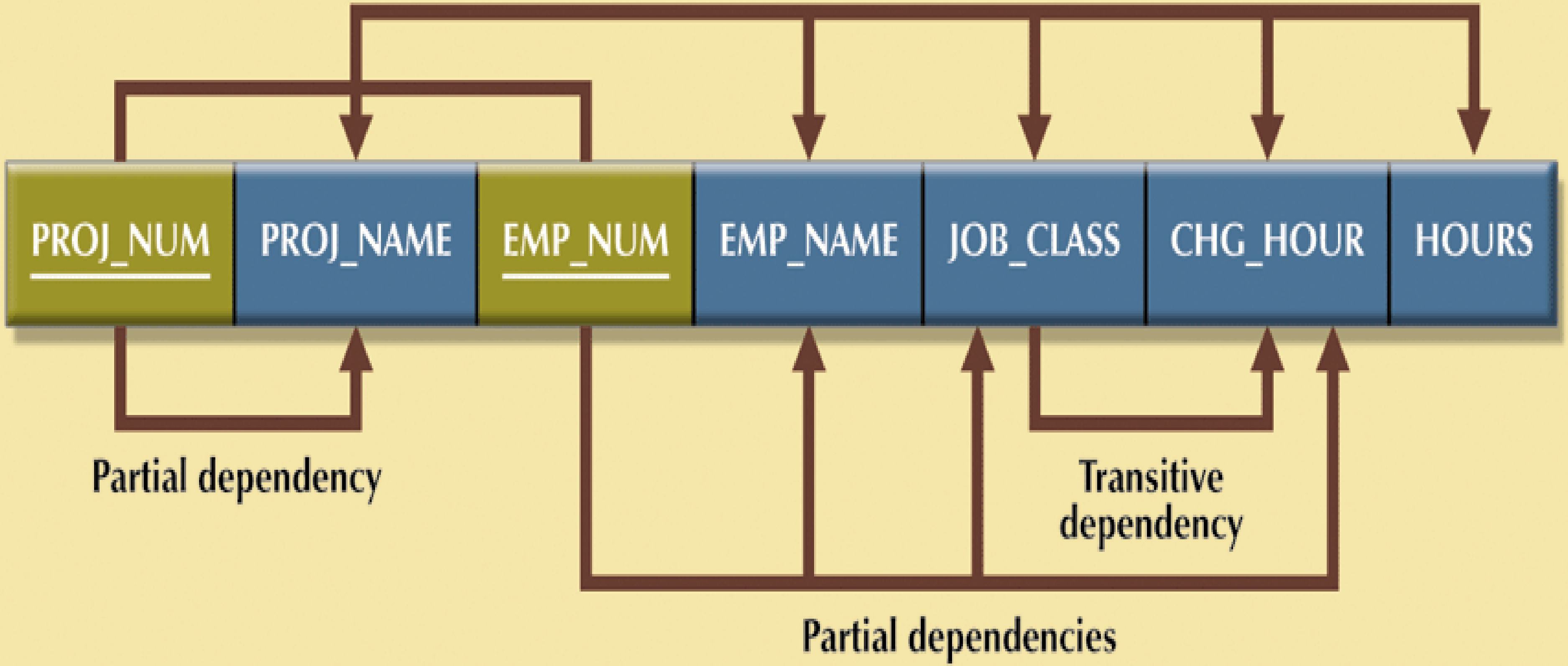
- **Third Normal Form**
  - Remove Transitive Dependencies.
  - Eliminate fields that do not depend on the key.
  - Search for another attribute to identify better than the key attribute.

Normalization

- **Third Normal Form**
  - EMPLOYEE (Emp_num, Emp_name, Job_class, Chg_hour)
  - ASSIGN (Emp_num, Proj_num, Hours)
    - EMPLOYEE (Emp_num, Emp_name, Job_class)
    - JOB (Job_class, Chg_hour)

Normalization

1st NF	Convert to Table Format, No repeating fields, Define Key attributes.
2nd NF	1st NF Eliminate partial dependencies
3rd NF	2nd NF Eliminate transitive dependencies



1NF (PROJ_NUM, EMP_NUM, PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOURS, HOURS)

PARTIAL DEPENDENCIES:
(PROJ_NUM → PROJ_NAME)
(EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR)

TRANSITIVE DEPENDENCY:
(JOB CLASS → CHG_HOUR)