SWE 207
Database Management Systems

~ Relational Model ~

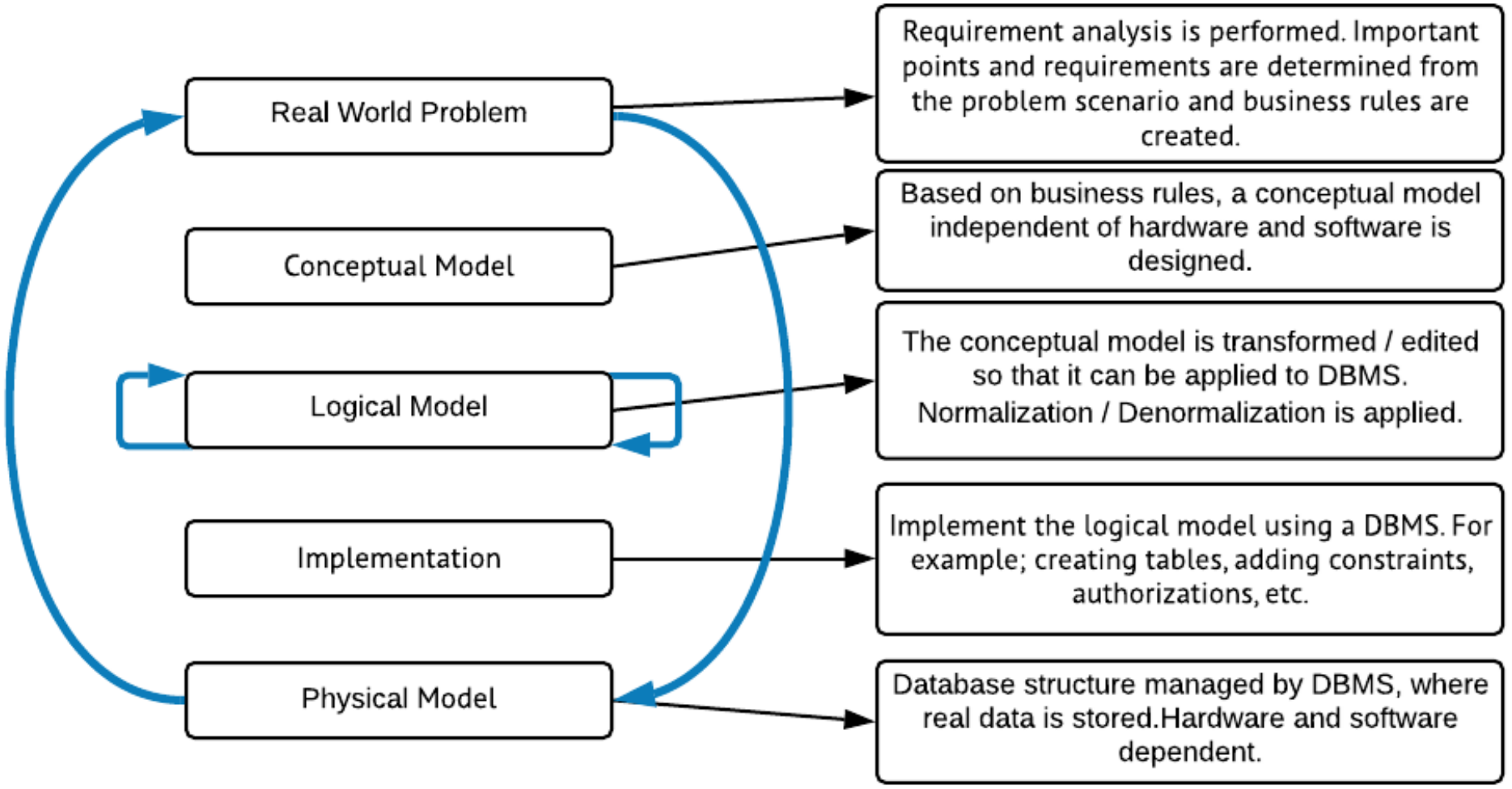
CELAL ÇEKEN,
GÖZDE YOLCU ÖZTEL,
VEYSEL HARUN ŞAHİN

SAKARYA
ÜNİVERSİTESİ

Content

- DB development life cycle
- Basic concepts
- Table
- Key
- Integrity rules
- Primary key characteristics
- Surrogate key
- Converting ERM to Relational Model
- Index
- Data dictionary
- Applications
  - Vehicle Producer
  - Electronic Commerce System

Database development lifecycle



Basic concepts

classId	catologId	term	facultyMember	branch
1	M001	17G	21	A
2	T002	18G	56	A
3	C001	17B	75	A
4	M001	18G	67	B
5	C001	18B	75	B

- Entity: Everything is that can be distinguished from existing and similar ones.
- Entity set: It is called the set of similar entities. For example, Students, Female Students, Dormitory Students, Doctors etc.
- Attribute: It is used to show the properties of objects in an entity set and to distinguish entities from each other. It corresponds to the "columns" in the table.
- Domain: It determines the values that an attribute can take.

Basic concepts

classId	catologId	term	facultyMember	branch
1	M001	17G	21	A
2	T002	18G	56	A
3	C001	17B	75	A
4	M001	18G	67	B
5	C001	18B	75	B

- Relation: It is a table constructed of rows and columns. Relation = Table
- Record (Tuple): They are records or rows in a table.
- In ERM⇒ Entity; In OOP⇒Object
  - A relation (table) can be thought of as rows or a set of records (all rows are different).
- Number of Columns (Degree): The number of columns in the relation (in the table).
- Number of Rows (Cardinality): The number of rows (records) in the relation (table).
- Primary Key: Uniq identifier. (not null, uniq)

Basic concepts

- Relational Database: Set of relations (tables).

A collection of tables made up of rows and columns.

- Relational Schema: It specifies the name of the relation (table), with the name and type of each column. It can be expressed in text or shown using a diagram.
  - For example, the relational schema (textually) of a relation (table) named Students is given below.
  - Students (studentID: string, firstName: string, lastName: string, age: integer, average: real)

Table

- Table Characteristics
  - A table is **perceived** as a two-dimensional structure consisting of rows and columns (data is shown being stored in table)
  - Actually in DBMS the data is stored in classical files. But it is shown like a table to the user. Tables improve the usability and make simpler the structure.
  - Because of that, the relational model is called a logical model, not a real model. The real model is classical file structers.
  - **Each table row** (record / tuple) represents **a single entity formation** within an entity set.
  - Each table column represents an attribute and each column has a **different name**.
  - Each row-column intersection represents a single data value.
  - A**ll values in a column** must have the **same data type.**
  - Each column has a precise range of values known as an attribute domain.
  - Each table must have an attribute or combination of attributes that uniquely identifies each row. This is called the **primary key**.

Table

customer_id	store_id	first_name	last_name	email	address_id	activebool	create_date	last_update	active
1	1	Mary	Smith	mary.smith@sakilacustomer.org	5	✓	2006-02-14	2013-05-26 14:49:45.738	1
2	1	Patricia	Johnson	patricia.johnson@sakilacustomer.org	6	✓	2006-02-14	2013-05-26 14:49:45.738	1
3	1	Linda	Williams	linda.williams@sakilacustomer.org	7	✓	2006-02-14	2013-05-26 14:49:45.738	1
4	2	Barbara	Jones	barbara.jones@sakilacustomer.org	8	✓	2006-02-14	2013-05-26 14:49:45.738	1
5	1	Elizabeth	Brown	elizabeth.brown@sakilacustomer.org	9	✓	2006-02-14	2013-05-26 14:49:45.738	1
6	2	Jennifer	Davis	jennifer.davis@sakilacustomer.org	10	✓	2006-02-14	2013-05-26 14:49:45.738	1
7	1	Maria	Miller	maria.miller@sakilacustomer.org	11	✓	2006-02-14	2013-05-26 14:49:45.738	1
8	2	Susan	Wilson	susan.wilson@sakilacustomer.org	12	✓	2006-02-14	2013-05-26 14:49:45.738	1
9	2	Margaret	Moore	margaret.moore@sakilacustomer.org	13	✓	2006-02-14	2013-05-26 14:49:45.738	1
10	1	Dorothy	Taylor	dorothy.taylor@sakilacustomer.org	14	✓	2006-02-14	2013-05-26 14:49:45.738	1
11	2	Lisa	Anderson	lisa.anderson@sakilacustomer.org	15	✓	2006-02-14	2013-05-26 14:49:45.738	1
12	1	Nancy	Thomas	nancy.thomas@sakilacustomer.org	16	✓	2006-02-14	2013-05-26 14:49:45.738	1
13	2	Karen	Jackson	karen.jackson@sakilacustomer.org	17	✓	2006-02-14	2013-05-26 14:49:45.738	1
14	2	Betty	White	betty.white@sakilacustomer.org	18	✓	2006-02-14	2013-05-26 14:49:45.738	1
15	1	Helen	Harris	helen.harris@sakilacustomer.org	19	✓	2006-02-14	2013-05-26 14:49:45.738	1
16	2	Sandra	Martin	sandra.martin@sakilacustomer.org	20	✓	2006-02-14	2013-05-26 14:49:45.738	0

Table

- The customer table is seen as a two-dimensional structure consisting of 16 rows (records) and 10 columns (attributes).
  - For example, first row identifies a customer named Mary Smith.
- Each column represents an attribute and each column has a different name.
- All values in a column are compatible with the attribute's characteristics.
- For example, the email column contains email information for all rows.

Table

- The data should be classified according to their form and function. Basic data types supported by DBMS:
  - Numeric: It is a data type on which meaningful arithmetic operations can be performed. For example, an area where the GPA is stored is of a numerical type.
  - Character: It can contain text data or a character string. For example, name, surname, phone number, etc. areas.
  - Date, Time: It is used for date and time data. For example; date of birth, time of last login, etc. areas.
  - Logical: True or false (yes or no - boolean) values.
- The range of allowed values for a column is called domain.
  - For example, since the student grade information in the table is limited to the range of 0-4, the value range of this column can be [0,4].

Key

- Functional Dependence
  - Key:
    - Records can be distinguished with a key (super, candidate, primary, secondary and surrogate keys)
    - A key that establishes relationships between tables (primary, foreign keys).
  - When determining the keys, the concept of functional dependency should be known.
  - In the example below, the customer name can be specified using the customerID field. The opposite is not true.
    - The customerID field determines the customerName field.
    - The customerName field is functionally dependent on the customerID field.
    - It is shown as customerID → customerName.

Table

customer_id	store_id	first_name	last_name	email	address_id	activebool	create_date	last_update	active
1	1	Mary	Smith	mary.smith@sakilacustomer.org	5	✓	2006-02-14	2013-05-26 14:49:45.738	1
2	1	Patricia	Johnson	patricia.johnson@sakilacustomer.org	6	✓	2006-02-14	2013-05-26 14:49:45.738	1
3	1	Linda	Williams	linda.williams@sakilacustomer.org	7	✓	2006-02-14	2013-05-26 14:49:45.738	1
4	2	Barbara	Jones	barbara.jones@sakilacustomer.org	8	✓	2006-02-14	2013-05-26 14:49:45.738	1
5	1	Elizabeth	Brown	elizabeth.brown@sakilacustomer.org	9	✓	2006-02-14	2013-05-26 14:49:45.738	1
6	2	Jennifer	Davis	jennifer.davis@sakilacustomer.org	10	✓	2006-02-14	2013-05-26 14:49:45.738	1
7	1	Maria	Miller	maria.miller@sakilacustomer.org	11	✓	2006-02-14	2013-05-26 14:49:45.738	1
8	2	Susan	Wilson	susan.wilson@sakilacustomer.org	12	✓	2006-02-14	2013-05-26 14:49:45.738	1
9	2	Margaret	Moore	margaret.moore@sakilacustomer.org	13	✓	2006-02-14	2013-05-26 14:49:45.738	1
10	1	Dorothy	Taylor	dorothy.taylor@sakilacustomer.org	14	✓	2006-02-14	2013-05-26 14:49:45.738	1
11	2	Lisa	Anderson	lisa.anderson@sakilacustomer.org	15	✓	2006-02-14	2013-05-26 14:49:45.738	1
12	1	Nancy	Thomas	nancy.thomas@sakilacustomer.org	16	✓	2006-02-14	2013-05-26 14:49:45.738	1
13	2	Karen	Jackson	karen.jackson@sakilacustomer.org	17	✓	2006-02-14	2013-05-26 14:49:45.738	1
14	2	Betty	White	betty.white@sakilacustomer.org	18	✓	2006-02-14	2013-05-26 14:49:45.738	1
15	1	Helen	Harris	helen.harris@sakilacustomer.org	19	✓	2006-02-14	2013-05-26 14:49:45.738	1
16	2	Sandra	Martin	sandra.martin@sakilacustomer.org	20	✓	2006-02-14	2013-05-26 14:49:45.738	0

Key

- Key Types
  - Super Key: The key that can uniquely identify each row is called a super key. It can contain one or more fields.
    - studentID
    - studentId, name
    - studentId, name, surname
    - e-mail
  - Candidate Key: In the super keys, the key which has the least number of field is called as candidate key.
    - "StudentID, name" is a super key. Only studentID field can uniquely identify records. So, "studentID, name" is not a candidate key.

Key

- Key Types
  - Primary Key, PK: It can define each row uniquely and it is selected from candidate keys.
    - The primary key must have unique values, it cannot take NULL values.
    - StudentID and passportID fields are candidate keys. Each can be selected as a primary key (only one must be chosen).
  - Alternative Key: Other candidate keys that are not selected as primary key are called secondary / alternative keys.
  - Surrogate Primary Key: In practice, surrogate primary keys are mostly used. Details are provided in the section titled " Surrogate Primary Key".

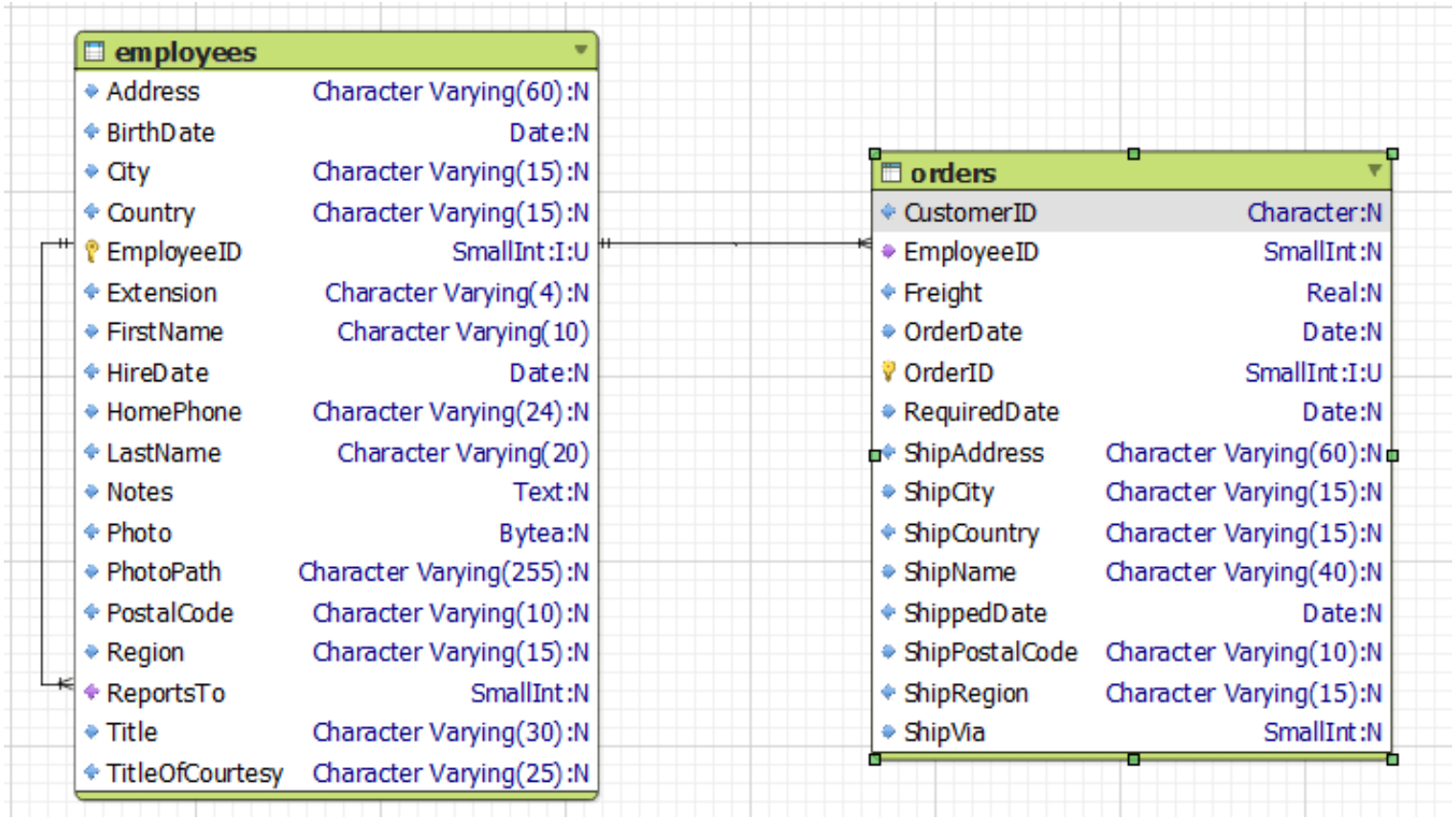
Key

- Key Types
  - Foreign Key, FK: A foreign key is a column in a table that refers to the primary key of another table. Thus, a relationship is established between the two tables.
    - When establishing a relationship between the two tables, the primary key of a table and the foreign key of another table is used. Therefore, both keys must be of the same type.
  - Below is an example of a foreign key.

Key

- Key Types
  - Foreign Key, FK:

An employee is responsible for many orders.
An order is the responsibility of an employee



Integrity rules

Integrity rules allow storing data correctly.

Owing to integrity rules, errors are prevented from begining.

If the error is prevented earlier, the error cost will be lower.

- Entity Integrity: Each table must have a primary key (PK) to provide entity integrity.
  - Each entity is guaranteed to have a unique ("unique" constraint)
  - For example: using entity integrity, it is guaranteed that
    - it is prevented to store multiple students with the same student number.
    - it is prevented to record students without a student number

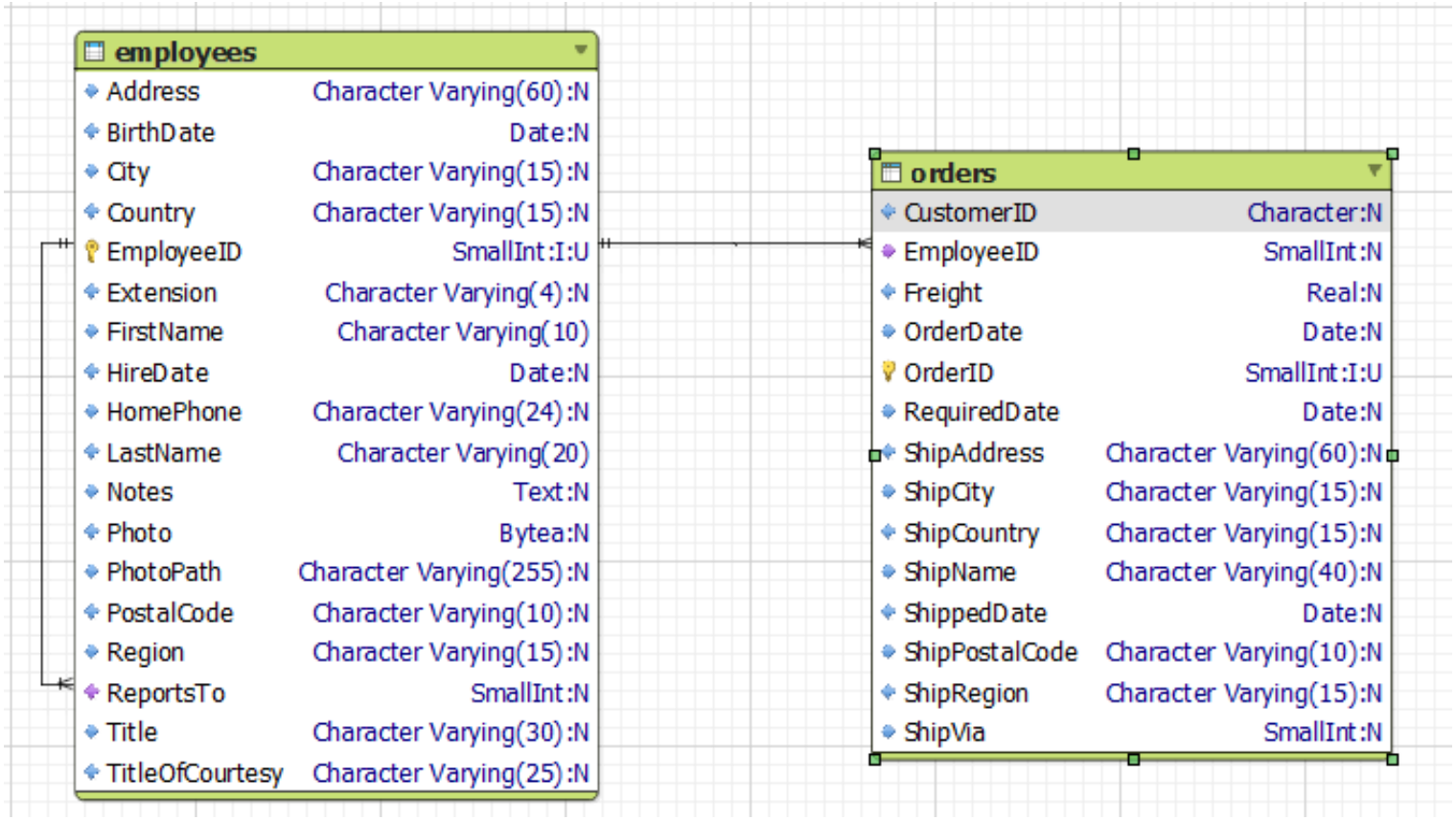
Integrity rules

- Referential Integrity: A foreign key (FK) is defined to provide reference integrity.
  - The foreign key field can be either NULL or one of the primary key field values in the other table.
  - Each foreign key value is guaranteed to be consistent with the primary key value of the other table.
  - For example: using reference integrity,
    - Assigning an order to a nonexistent employeer is prevented.
    - a student taking an unopened course is prevented.
    - deleting a student enrolled in the course is prevented.
    - ordering a nonexistent product, etc., is prevented.

Key

- Key Types
  - Foreign Key, FK:

An employee is responsible for many orders.
An order is the responsibility of an employee



Key

- Key Types
  - Foreign Key, FK:

Database: NorthWind Schema: public Table: employees Revert Tables Layout: Default

	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	Address	City	Region	PostalCode	Country	HomePhone	Extension
1	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08	1992-05-01	507 - 20th Ave. E.\nApt. 2A	Seattle	WA	98122	USA	(206) 555-9857	5467
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19	1992-08-14	908 W. Capital Way	Tacoma	WA	98401	USA	(206) 555-9482	3457
3	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30	1992-04-01	722 Moss Bay Blvd.	Kirkland	WA	98033	USA	(206) 555-3412	3355
4	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19	1993-05-03	4110 Old Redmond Rd.	Redmond	WA	98052	USA	(206) 555-8122	5176
5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04	1993-10-17	14 Garrett Hill	London	<NULL>	SW1 8JR	UK	(71) 555-4848	3453
6	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02	1993-10-17	Coventry House\nMiner Rd.	London	<NULL>	EC2 7JR	UK	(71) 555-7773	428
7	7	King	Robert	Sales Representative	Mr.	1960-05-29	1994-01-02	Edgeham Hollow\nWinchester Way	London	<NULL>	RG1 9SP	UK	(71) 555-5598	465
8	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09	1994-03-05	4726 - 11th Ave. N.E.	Seattle	WA	98105	USA	(206) 555-1189	2344
9	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27	1994-11-15	7 Houndstooth Rd.	London	<NULL>	WG2 7LT	UK	(71) 555-4444	452

9 from 9

Related Table: orders : lnk_employees_orders (1:M) Mode: show linked Set: OR (union) Auto

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName	ShipAddress	ShipCity	ShipRegion	ShipPostal
1	10248	VINET	5	1996-07-04	1996-08-01	1996-07-16	3	32.38	Vins et alcools Chevalier	59 rue de l'Abbaye	Reims	<NULL>	51100
2	10249	TOMSP	6	1996-07-05	1996-08-16	1996-07-10	1	11.61	Toms Spezialitäten	Luisenstr. 48	Münster	<NULL>	44087
3	10250	HANAR	4	1996-07-08	1996-08-05	1996-07-12	2	65.83	Hanari Carnes	Rua do Paço, 67	Rio de Janeiro	RJ	05454-876
4	10251	VICTE	3	1996-07-08	1996-08-05	1996-07-15	1	41.34	Victuailles en stock	2, rue du Commerce	Lyon	<NULL>	69004
5	10252	SUPRD	4	1996-07-09	1996-08-06	1996-07-11	2	51.3	Suprêmes délices	Boulevard Tirou, 255	Charleroi	<NULL>	B-6000
6	10253	HANAR	3	1996-07-10	1996-07-24	1996-07-16	2	58.17	Hanari Carnes	Rua do Paço, 67	Rio de Janeiro	RJ	05454-876
7	10254	CHOPS	5	1996-07-11	1996-08-08	1996-07-23	2	22.98	Chop-suey Chinese	Hauptstr. 31	Bern	<NULL>	3012
8	10255	RICSU	9	1996-07-12	1996-08-09	1996-07-15	3	148.33	Richter Supermarkt	Starenweg 5	Genève	<NULL>	1204
9	10256	WELLI	3	1996-07-15	1996-08-12	1996-07-17	2	13.97	Wellington Importadora	Rua do Mercado, 12	Resende	SP	08737-363
10	10257	HILAA	4	1996-07-16	1996-08-13	1996-07-22	3	81.91	HILARION-Abastos	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022
11	10258	ERNSH	1	1996-07-17	1996-08-14	1996-07-23	1	140.51	Ernst Handel	Kirchgasse 6	Graz	<NULL>	8010
12	10259	CENTC	4	1996-07-18	1996-08-15	1996-07-25	3	3.25	Centro comercial Moctezuma	Sierras de Granada 9993	México D.F.	<NULL>	05022

Integrity rules

- Referential integrity notes:
  - It was also said that the foreign key field could be NULL to provide referential integrity.
  - Since NULL values can cause problems in queries, special values are generally preferred over NULL values.
    - For example, "00: Unknown" can be given instead of NULL as province information. In this case, there should be a row with 00, Unknown values in the Provinces table. The default value of the province field in the contacts table can also be set to 00. Thus, unknown information is assigned by default when the province information of the person is not entered.

Integrity rules

- NOT NULL: If we want to force a value into a field, NOT NULL can be used.
  - For ex: We want to prevent inserting a student without his name. So we need to define that area as not null.

- UNIQUE : It is used if we want the values written in the field to be unique for each row.
  - For example, we guarantee that students' student numbers cannot be written the same.

Primary key characteristics

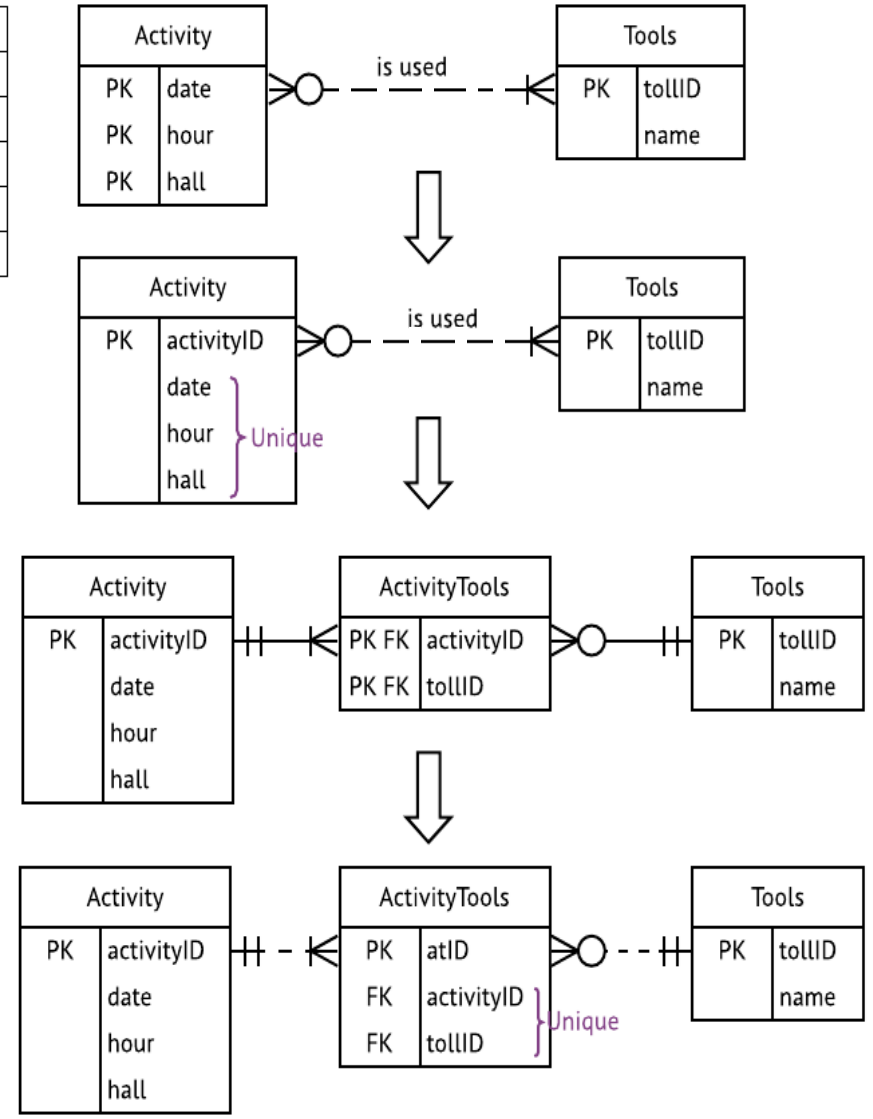
- Provide UNIQUE (uniquely identify records) and NOT NULL integrity rules.
- Fields that are likely to change over time are not preferred.
- It should consist of a single field as much as possible.
  - If it consists of many fields, it becomes complicated to establish relationships and write queries. Latency and resource utilization increase.
- Numerical fields should be preferred.
  - Processing speed and auto-increment support can be provided.
- Fields with security risks should not be preferred.
  - The primary key value is used extensively in software.

Surrogate key

- It is the field added to perform primary key functions.

Ex: In a congress center there are many conferance halls, and in these halls many activities are organized.

date	hour	hall
1.1.2000	10.00	1
1.1.2000	11.00	1
2.1.2000	10.00	1
2.1.2000	11.00	1
2.1.2000	11.00	2



Converting ERM to Relational Model

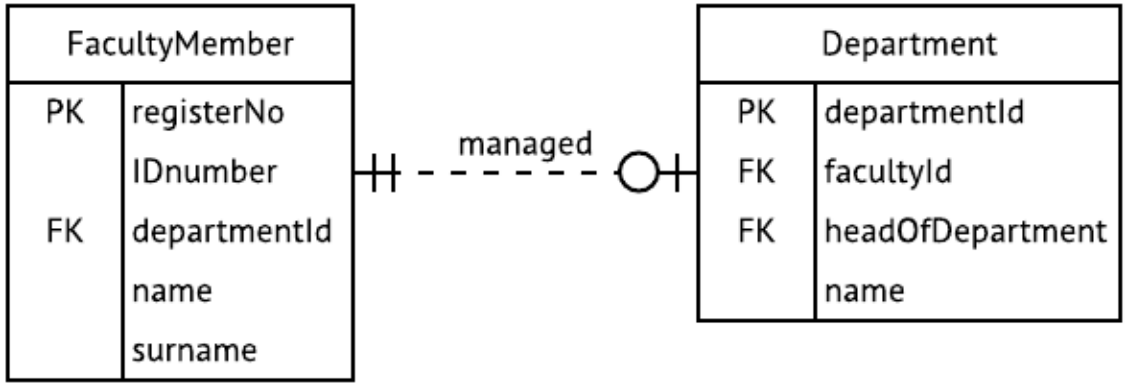
- Entities are tranformed to a table.

Student	
PK	studentId
FK	departmentId
	name
	surname

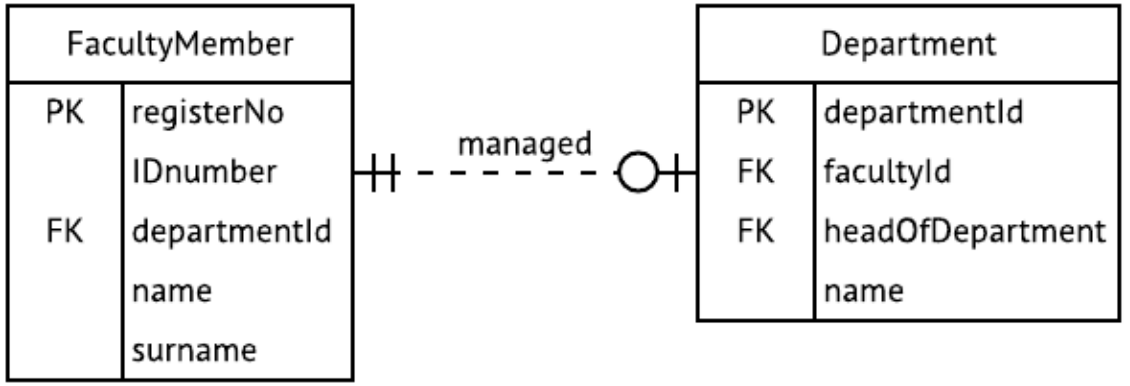
- Relational Schema (Textual Representation)
  - Student(<u>studentId: String</u>, departmentId: Integer, name: String, surname: String)

Converting ERM to Relational Model

- <u>One to One Relationship:</u>
  - 1 faculty member can manage at most 1 department.
  - 1 department can be managed by only 1 faculty member.



Converting ERM to Relational Model

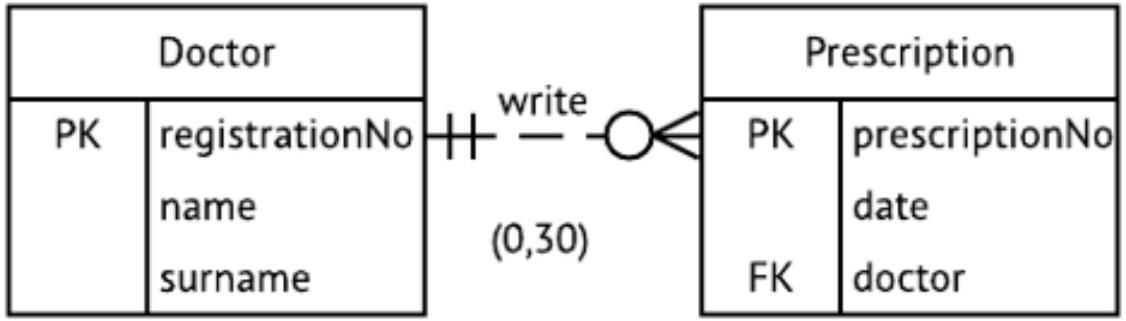


- One to One Relationship:

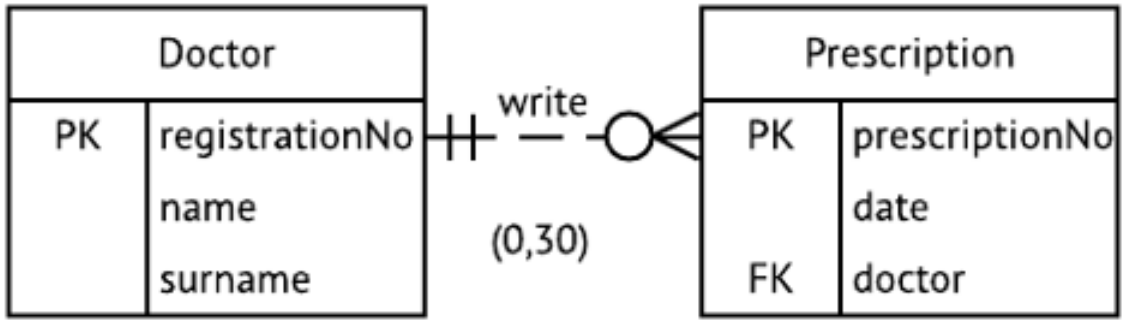
  - The primary key (registerNo) of the "only 1" side is added to the other entity as a foreign key field (headOfDepartment).
  - Relational Schema (Textual Representation) :
    - FacultyMember(registerNo: String, IDnumber: Char(11), departmentId: Integer, name: String, surname: String)
    - Department(departmentId: Integer, facultyId: Integer, headOfDepartment: String, name: String)

Converting ERM to Relational Model

- <u>One to Many Relationship:</u>
  - 1 prescription is written by one doctor.
  - 1 doctor can write many prescription.



Converting ERM to Relational Model

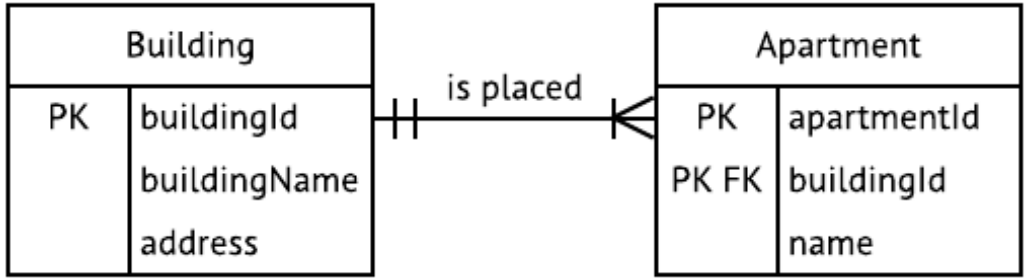


- One to Many Relationship:
  - The primary key (registrationNo) of the "only 1" side is added to the other entity as a foreign key field (doctor).
  - Relational Schema (Textual Representation) :
    - Doctor(registrationNo: Integer, name: String, surname)
    - Prescription(prescriptionNo:Integer, date:Date, doctor: Integer)

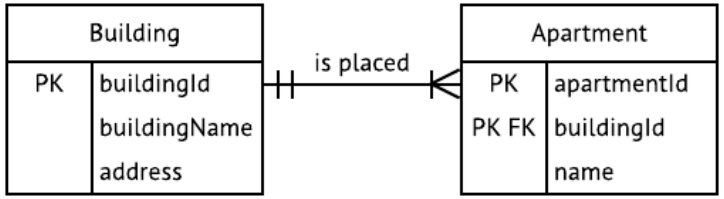
Converting ERM to Relational Model

- <u>One to Many Relationship / Existence Dependency, Relationship of Definition:</u>

  - There are many apartments in 1 building. There must be at least one.
  - 1 apartment must be in only 1 building.



Converting ERM to Relational Model

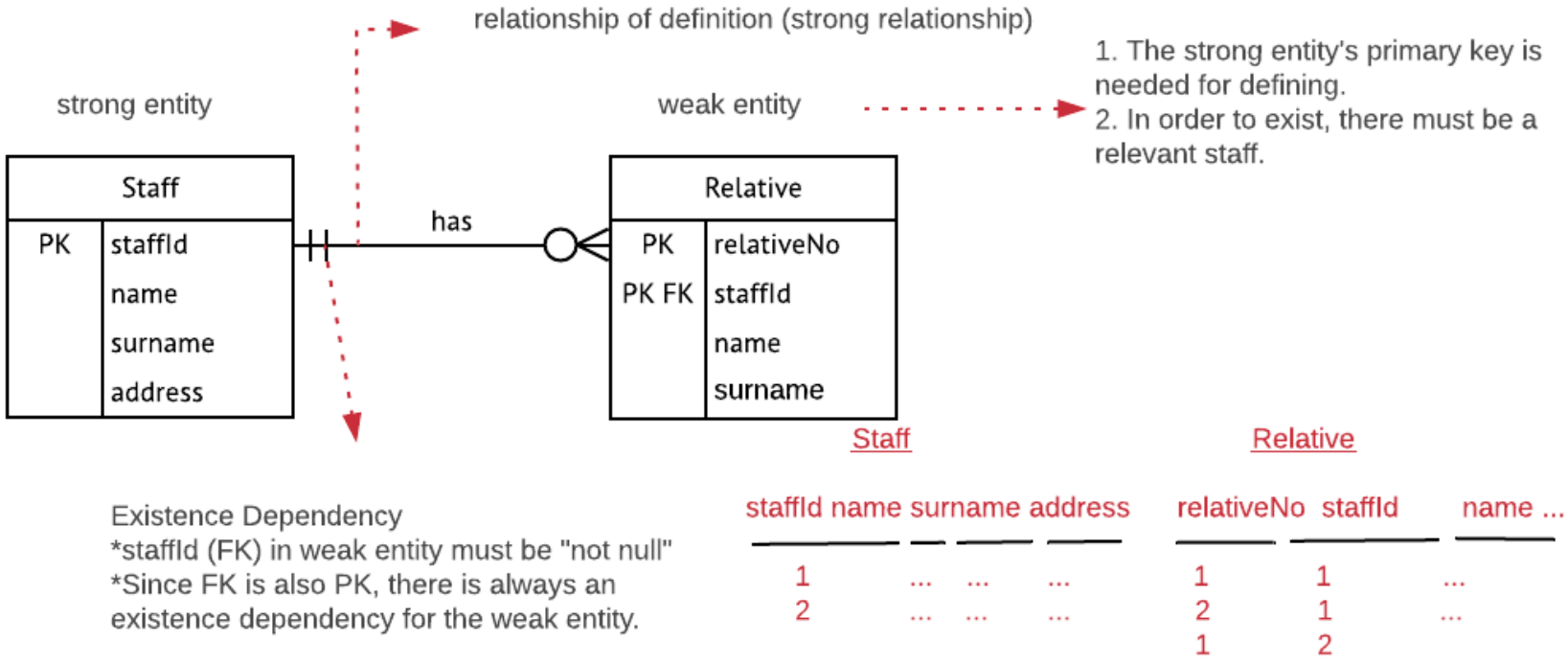


- One to Many Relationship / Existence Dependency, Relationship of Definition:

  - The primary key (buildingId) of the "only 1" side is added to the other entity as a foreign key and primary key field (buildingId).
  - Relational Schema (Textual Representation) :
    - Building(buildingId: Integer, buildingName: String, address: String)
    - Apartment(apartmentId: Integer, buildingId: Integer, name: String)

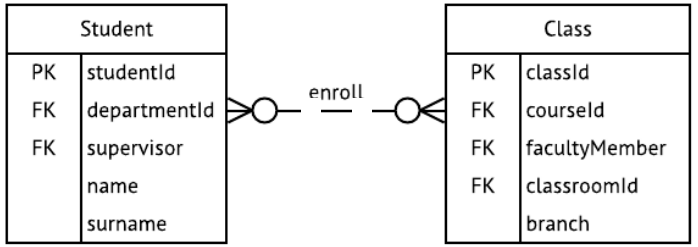
Converting ERM to Relational Model

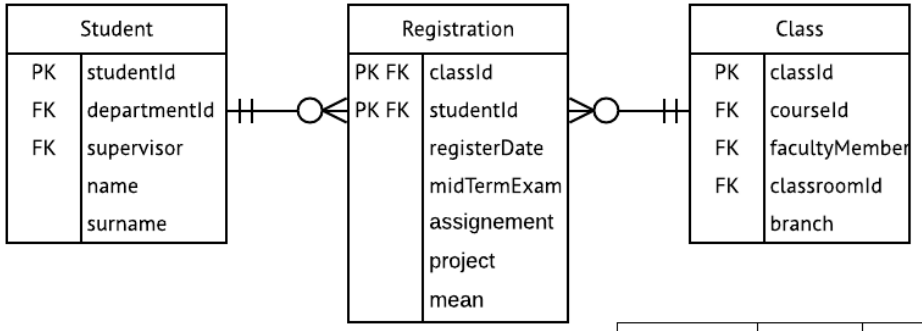
- One to Many Relationship / Existence Dependency, Relationship of Defini



Converting ERM to Relational Model

- Many to Many Relationship:

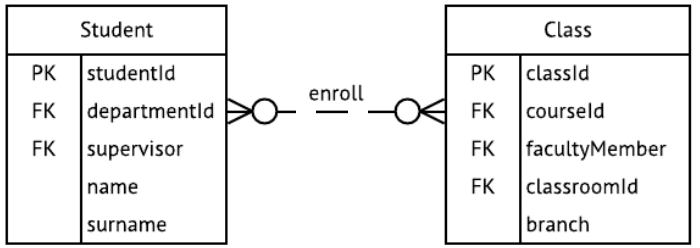




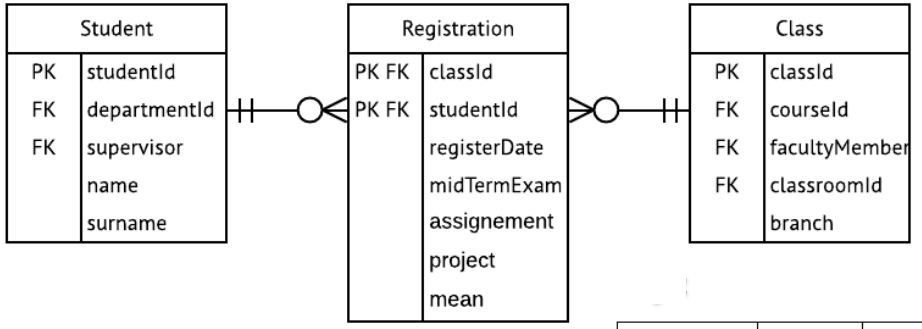
classId	studentId	midTermExam	assignement	project ...
1	1	80	21	100
1	2	90	56	90
3	1	90	75	40
3	2	50	67	95
3	3	40	75	100

- 1 student can enroll in many classes.
- 1 class can be taken by many students

Converting ERM to Relational Model



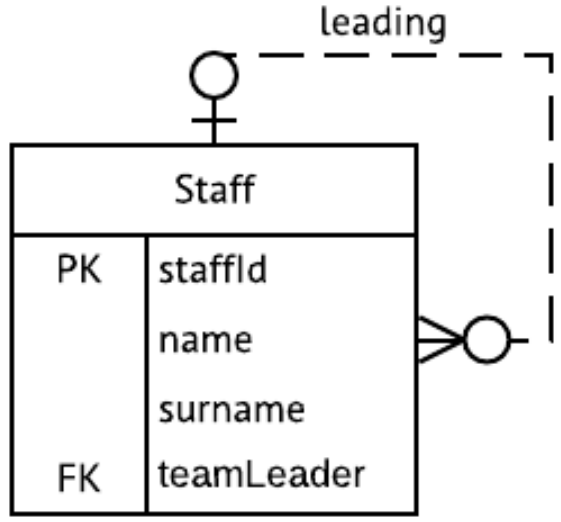
- Many to Many Relationship:



classId	studentId	midTermExam	assignement	project ...
1	1	80	21	100
1	2	90	56	90
3	1	90	75	40
3	2	50	67	95
3	3	40	75	100

- Relational Schema (Textual Representation) :
  - Student(studentId: String, departmentId: Integer, supervisor: String, name: String, surname: String)
  - Registration(classId: String, studentId: String, registerDate: Date, …)
  - Class(classId: String, courseId: String, facultyMember: String, classroomId: Integer, branch: Char(1))

Converting ERM to Relational Model

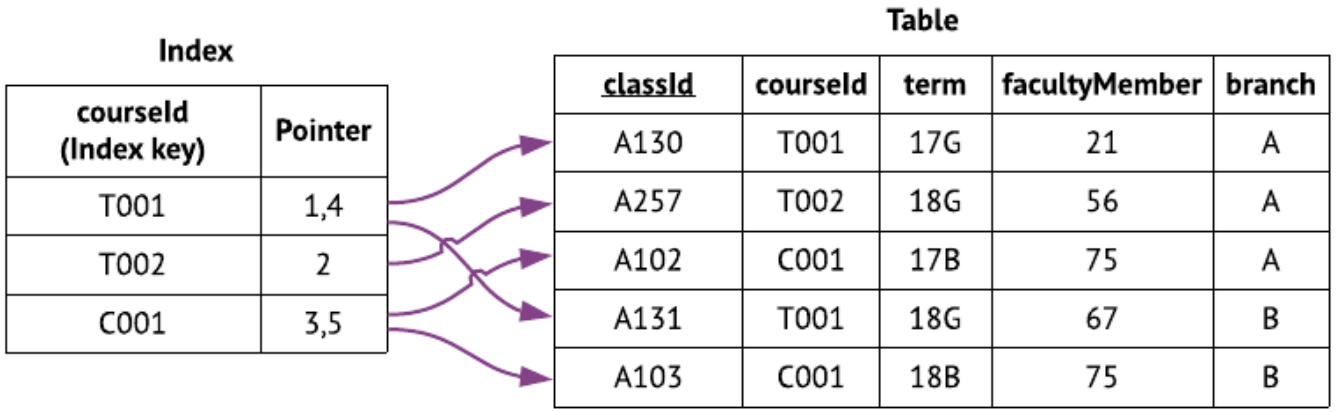


- Unary relationship:
  - Only 1 staff leads 1 staff.
  - 1 staff leads many staff.

  - Relational Schema (Textual Representation) :
    - Staff(staffId: String, name: String, surname: String, teamLeader: String)

Index

- An index is used for increasing the search and list speed.
- The location of the searched information is found using the key value.
- Primary key fields automatically become unique index.
-



Index

courseId (Index key)	Pointer
T001	1,4
T002	2
C001	3,5

Table

classId	courseId	term	facultyMember	branch
A130	T001	17G	21	A
A257	T002	18G	56	A
A102	C001	17B	75	A
A131	T001	18G	67	B
A103	C001	18B	75	B

Index

- An index can be destroyed or created without affecting other indexes.
- Generally, the features that searches frequently are made index.
- Searches with index fields are very fast, while record insertion and deletion (INSERT, DELETE) operations are relatively slow. Because the index table have to be updated acording to insert delete update operations.
- Index fields are listed sequentially.
- Multiple indexes can be defined in a table. Each index can belong to only one table.

Data Dictionary

- It is a database where metadata is stored.
- It is managed by the Database Management System.
- Metadata of all the databases (tables, fields, field types, value ranges, keys, indexes, relationships, constraints, etc.) are stored here.
- The user can query it.

Vehicle Producer

- Scenario
  - A vehicle manufacturer produces different models of vehicles.
  - The manufacturer demands a software system.
  - The manufacturer wants to store and manage some information like the vehicles, the materials used in the vehicles, suppliers, etc.

Vehicle Producer

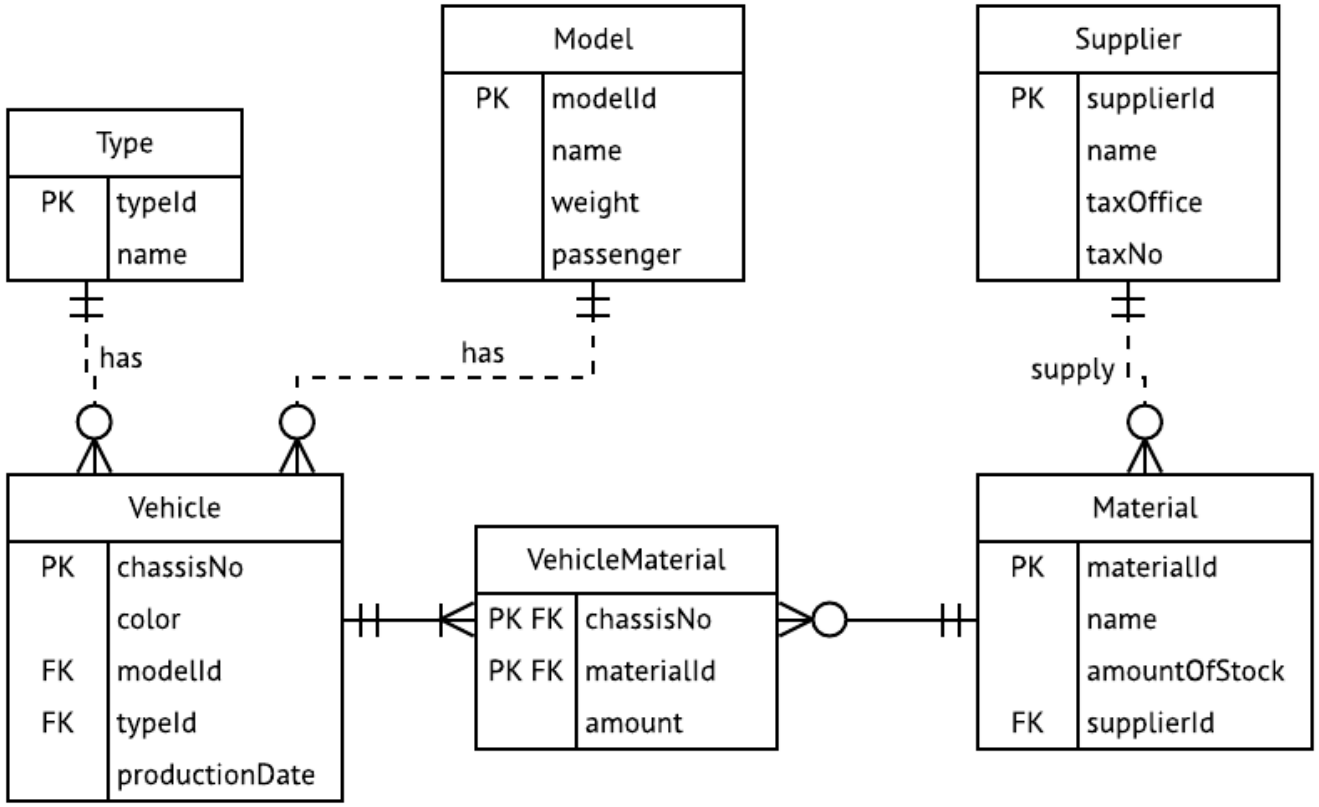
- Business Rules
  - Each vehicle has chassis number, color, type (automobile, truck, minibus, etc.), model and production date information. Vehicles are distinguished from each other by their chassis number.
  - The vehicle types have a code and name.
  - Vehicle models have a code, name, weight and number of passengers information.
  - Materials have a code, name and amount of stock information.
  - Materials are purchased from supplier companies.
  - Supplier companies have a code, name, tax office, tax number information.

Vehicle Producer

- Business Rules
  - A supplier may not supply any material or supply one or more materials. A material is supplied only by one supplier.
  - A vehicle consists of many materials. It must include at least one material. One material can be used in more than one vehicle. However, it may not have been used in any vehicle yet.
  - Vehicles are made of different amounts and types of materials. This information should also be recorded.
  - A vehicle can only have one type. There may be no vehicle of a type or many vehicles of a type.
  - A vehicle has only one model. There may not be any vehicle belonging to a model, or there may be many vehicles belonging to a model.

Vehicle Producer

- Entity Relationship Model



Vehicle Producer

- Relational Schema (Textual Representation):
  - Vehicle(chassisNo:char, color: varchar, modelId: char, typeId: char, productionDate: date)
  - VehicleMaterial(chassisNo : char, materialId: char, amount: int)
  - Material(materiald : char, name: varchar, amountOfStock: int, supplierId: char)
  - Type(typeId: char, name: varchar)
  - Model(modelId: char, name: varchar, weight: real, passenger: int)
  - Supplier(supplierId : char, name: varchar, taxOffice: varchar, taxNo: char)

Electronic Commerce System

- Scenario
  - An electronic commerce company is requested to design a database for an electronic commerce system. In the designed database; customers, products, orders, receipts, sales representatives, etc. information is expected to store.

Electronic Commerce System

- Business Rules:
  - In this database, a number is given to each customer and the customer's IDnumber, name, surname, and province information are planned to be stored.
  - The license plate number and name of the provinces are kept.
  - There is a sales representative who is interested in every order. IDnumber, name and surname of the sales representatives are available.
  - The code, name, price and stock quantities of the products (PC, telephone, etc.) must be stored.
  - Categories of products have category id and category name(computer, home electronics, cosmetics, etc.) information.

Electronic Commerce System

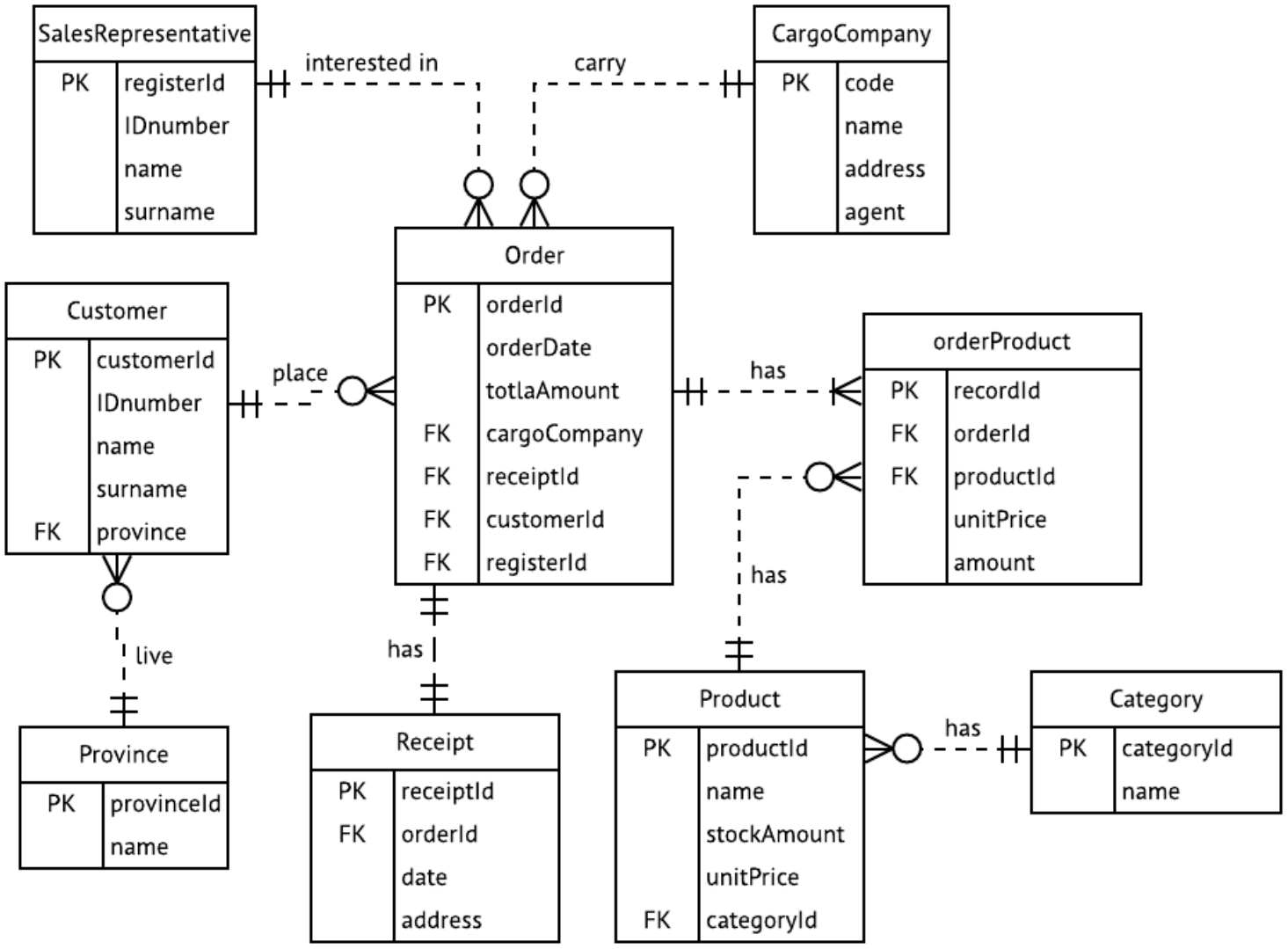
- Business Rules:
  - Customers' orders should be stored and a receipt should be available for each order. Each order has a unique order number and order date.
  - The receipt number, date and billing address information of the receipts should be stored. (Total price can be calculated or stored).
  - Orders are carried by a cargo company. The cargo company has a code, name, address. Each cargo company has an officier responsible for orders.
  - A product has only one category. A category can have multiple products or no product.

Electronic Commerce System

- Business Rules:
  - An order contains at least one product but can also contain multiple products. A product can take place in multiple orders or no order.
  - A customer can place many orders or no order. An order can only be placed by just one customer.
  - An order can have only one receipt. A receipt can only be for one order.
  - Only one sales representative is interested in an order. A sales representative can interest in many orders or no order.
  - A customer can live in only one province. Many customers can live in a province.
  - An order is carried by only one cargo company. A cargo company can carry many orders or no order.
  - The order quantity and unit price of an ordered product should also be recorded.

Electronic Commerce Syste

- Entity Relationship Model



Electronic Commerce System

- Relational Schema (Textual Representation):
  - SalesRepresentative(registerId: char, IDnumber: char, name: varchar, surname: varchar)
  - CargoCompany(code: char, name: varchar, address: varchar, agent: varchar)
  - Customer(customerId: char, IDnumber: char, name: varchar, surname: varchar, province: varchar)
  - Order(orderId: char, orderDate: date, totalAmount: real, cargoCompany: char, receiptId: char, customerId: char, registerId: char)
  - OrderProduct(recordId: int, orderId: char, productId: char, unitPrice: real, amount: int)
  - Category(categoryId: char, name: varchar)
  - Province(provinceId: varchar, name: varchar)
  - Receipt(receiptId: char, date: date)
  - Product(productId: char, name: varchar, stockAmount: int, unitPrice: real, categoryId: char)