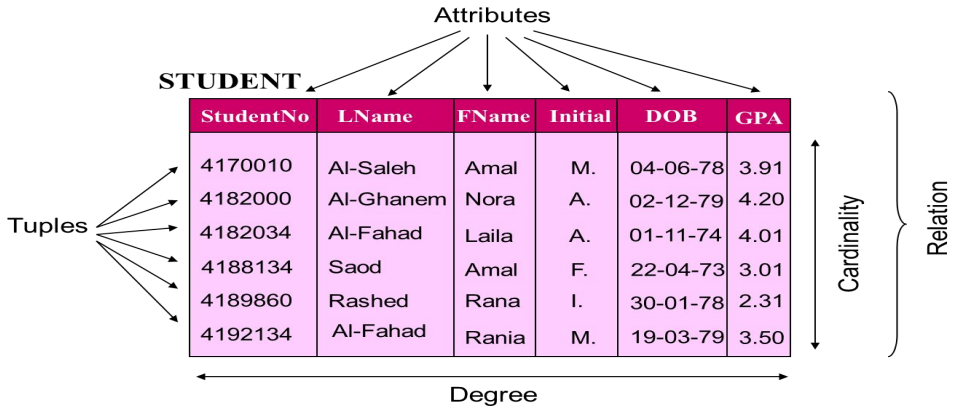
Lab Content:

- Relational model and relational mapping
  (Mapping for Students' Case Study)
- Introduction to SQL
  - What is SQL
  - SQL data types, nulls

Relational model and relational mapping

(Mapping for Students' Case Study)



- **Relation** is a table with columns & rows. Holds information about entities.
- **Attribute** is a named column of a relation.
- **Tuple** is a row of a relation.
- **Degree** of a relation is the number of attributes it contains.
- **Cardinality** of a relation is the number of tuples it contains.

ER ➔ Relational Model

Entity Type

- Represent each entity type with a relation.
- Entity type attributes become the relation attributes.

STUDENT (StudentNo, Lname, Fname, Initial, DOB, GPA, Dept)
DEPARTMENT (DeptNo, Department Name, Location)

Weak Entity Type

A weak entity type relation must include its key and its strong entity type PK as a FK. The combination of the two keys form the PK of the weak entity.

EMPLOYEE (EmpNo, Lname, Fname, DOB)
DEPENDENT (DepNo, EmpNo, FName)

1:1 Relationship

- Identify an entity type (S) (preferably total participator).
- Include the PK of the other entity (T) as a FK in S.
- Add attributes that describes the relationship to S.

EMPLOYEE(<u>EmpNo</u>, Lname, Fname, DOB)
BRANCH(<u>BrnNo</u>, Name, <u>EmpNo</u>, StartDate, EndDate)

1:M Relationship

- Identify a participating entity type (S) on the m-side.
- Include the PK of the other entity type (T) as a FK in S.
- Add attributes that describes the relationship to S.

EMPLOYEE(<u>EmpNo</u>, Lname, Fname, DOB, <u>BrnNo</u>)
BRANCH(<u>BrnNo</u>, Name)

M:N Relationship

- Create a relation R to represent the relationship.
- Include the PK of participating entity types (T & S) as FK in R. The combination of the two FK will form the PK of R.
- Add attributes that describes the relationship to R.

EMPLOYEE(<u>EmpNo</u>, Lname, Fname, DOB)
PROJECT(<u>ProjNo</u>, Name)
Work-on(<u>EmpNo</u>,<u>ProjNo</u>, hours)

n-ary Relationship

- Create a relation R to represent the relationship.
- Include the PK of the participating entities as FK in R. The combination of all FK form the PK of R.
- Add attributes that describes the relationship to R.

BUSINESS(<u>BizNo</u>) LAWYER(<u>LawNo</u>) SUPPLIER(<u>SupNo</u>)
contract(<u>BizNo</u>, <u>SupNo</u>, <u>LawNo</u>, StartDate, EndDate)

Composite Attribute

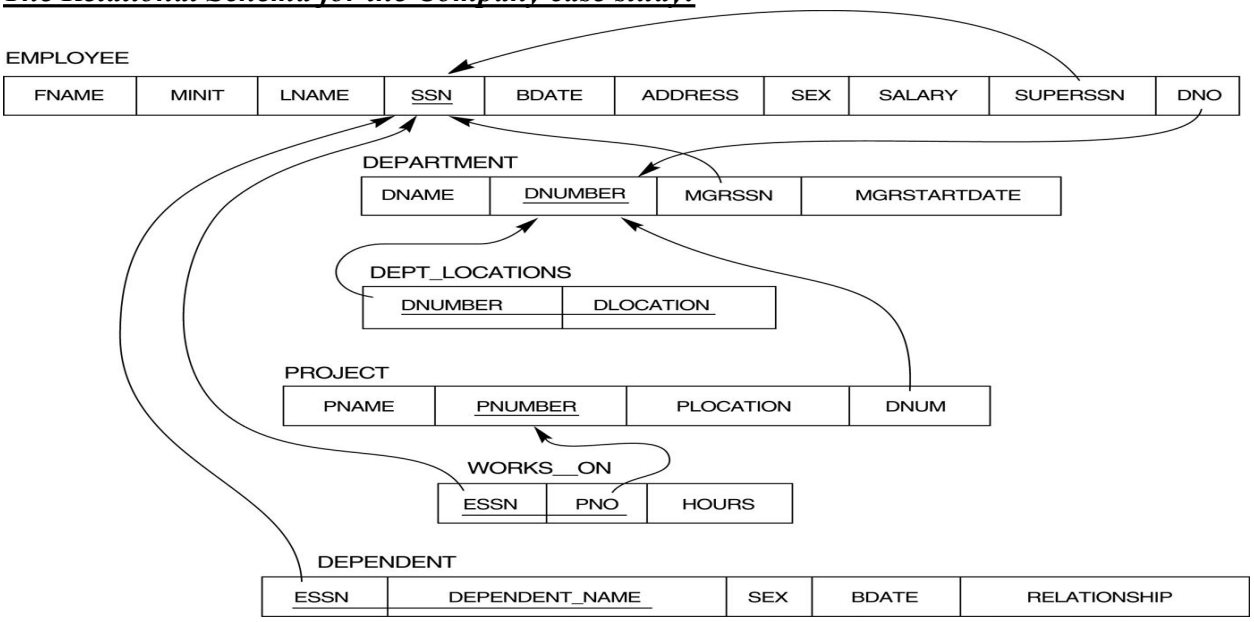
Include its simple components in the relation.
EMPLOYEE(<u>EmpNo</u>, Fname, initial, Lname, DOB)

MultiValue Attribute

- Suppose A is a relation that contains the multivalued attribute.
- Create a relation R to represent the attribute.
- Include the PK of A as FK in R.
- The PK of R is the combination of the PK of A (FK) & the multivalued attribute.

EMPLOYEE(<u>EmpNo</u>, DOB)
TELEPHONE(<u>EmpNo</u>, <u>tel_no</u>)

The Relational Schema for the Company case study:



Introduction to SQL

What is SQL

- SQL (pronounced "ess-que-el") stands for Structured Query Language.
- SQL is used to communicate with a database.
- According to ANSI (American National Standards Institute), it is the standard language for relational database management systems.
- SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.
- Some common relational database management systems that use SQL are:
    - Oracle,
    - Sybase,
    - Microsoft SQL Server,
    - Access,
    - Ingres, etc.
- Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system.

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

SQL data types, nulls

The following table lists the general data types in SQL:

Data type	Description
CHARACTER(n)	Character string. Fixed-length n
VARCHAR(n) or CHARACTER VARYING(n)	Character string. Variable length. Maximum length n
BINARY(n)	Binary string. Fixed-length n
BOOLEAN	Stores TRUE or FALSE values
VARBINARY(n) or BINARY VARYING(n)	Binary string. Variable length. Maximum length n
INTEGER(p)	Integer numerical (no decimal). Precision p
SMALLINT	Integer numerical (no decimal). Precision 5
INTEGER	Integer numerical (no decimal). Precision 10
BIGINT	Integer numerical (no decimal). Precision 19
DECIMAL(p,s)	Exact numerical, precision p, scale s. Example: decimal(5,2) is a number that has 3 digits before the decimal and 2 digits after the decimal
NUMERIC(p,s)	Exact numerical, precision p, scale s. (Same as DECIMAL)
FLOAT(p)	Approximate numerical, mantissa precision p. A floating number in base 10 exponential notation. The size argument for this type consists of a single number specifying the minimum precision
REAL	Approximate numerical, mantissa precision 7
FLOAT	Approximate numerical, mantissa precision 16
DOUBLE PRECISION	Approximate numerical, mantissa precision 16
DATE	Stores year, month, and day values
TIME	Stores hour, minute, and second values
TIMESTAMP	Stores year, month, day, hour, minute, and second values
INTERVAL	Composed of a number of integer fields, representing a period of time, depending on the type of interval

ARRAY	A set-length and ordered collection of elements
MULTISET	A variable-length and unordered collection of elements
XML	Stores XML data

SQL Server Data Types

String types:

Data type	Description	Storage
char(n)	Fixed width character string. Maximum 8,000 characters	Defined width
varchar(n)	Variable width character string. Maximum 8,000 characters	2 bytes + number of chars
varchar(max)	Variable width character string. Maximum 1,073,741,824 characters	2 bytes + number of chars
text	Variable width character string. Maximum 2GB of text data	4 bytes + number of chars
nchar	Fixed width Unicode string. Maximum 4,000 characters	Defined width x 2
nvarchar	Variable width Unicode string. Maximum 4,000 characters	
nvarchar(max)	Variable width Unicode string. Maximum 536,870,912 characters	
ntext	Variable width Unicode string. Maximum 2GB of text data	
bit	Allows 0, 1, or NULL	
binary(n)	Fixed width binary string. Maximum 8,000 bytes	
varbinary	Variable width binary string. Maximum 8,000 bytes	
varbinary(max)	Variable width binary string. Maximum 2GB	
image	Variable width binary string. Maximum 2GB	

Number types:

Data type	Description	Storage
tinyint	Allows whole numbers from 0 to 255	1 byte
smallint	Allows whole numbers between -32,768 and 32,767	2 bytes
int	Allows whole numbers between -2,147,483,648 and 2,147,483,647	4 bytes
bigint	Allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807	8 bytes
decimal(p,s)	Fixed precision and scale numbers. Allows numbers from -10^38 +1 to 10^38 –1. The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18. The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0	5-17 bytes
numeric(p,s)	Fixed precision and scale numbers. Allows numbers from -10^38 +1 to 10^38 –1. The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18. The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0	5-17 bytes
smallmoney	Monetary data from -214,748.3648 to 214,748.3647	4 bytes
money	Monetary data from -922,337,203,685,477.5808 to 922,337,203,685,477.5807	8 bytes
float(n)	Floating precision number data from -1.79E + 308 to 1.79E + 308. The n parameter indicates whether the field should hold 4 or 8 bytes. float(24) holds a 4-byte field and float(53) holds an 8-byte field. Default value of n is 53.	4 or 8 bytes
real	Floating precision number data from -3.40E + 38 to 3.40E + 38	4 bytes

Date types:

Data type	Description	Storage
datetime	From January 1, 1753 to December 31, 9999 with an accuracy of 3.33 milliseconds DATETIME - format: YYYY-MM-DD HH:MM:SS	8 bytes
datetime2	From January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds	6-8 bytes
smalldatetime	From January 1, 1900 to June 6, 2079 with an accuracy of 1 minute SMALLDATETIME - format: YYYY-MM-DD HH:MM:SS	4 bytes
date	Store a date only. From January 1, 0001 to December 31, 9999 DATE - format YYYY-MM-DD	3 bytes
time	Store a time only to an accuracy of 100 nanoseconds	3-5 bytes
datetimeoffset	The same as datetime2 with the addition of a time zone offset	8-10 bytes
timestamp	Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable TIMESTAMP - format: a unique number	

Other data types:

Data type	Description
sql_variant	Stores up to 8,000 bytes of data of various data types, except text, ntext, and timestamp
uniqueidentifier	Stores a globally unique identifier (GUID)
xml	Stores XML formatted data. Maximum 2GB
cursor	Stores a reference to a cursor used for database operations
table	Stores a result-set for later processing

SQL NULL Values

- NULL values represent missing unknown data.
- By default, a table column can hold NULL values.

SQL NULL Values

- If a column in a table is optional, we can insert a new record or update an existing record without adding a value to this column. This means that the field will be saved with a NULL value.
- NULL values are treated differently from other values.
- NULL is used as a placeholder for unknown or inapplicable values.

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola		Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari		Stavanger

Suppose that the "Address" column in the "Persons" table is optional. This means that if we insert a record with no value for the "Address" column, the "Address" column will be saved with a NULL value.
How to test for NULL values?
It is not possible to test for NULL values with comparison operators, such as =, <, or <>.
Use the IS NULL and IS NOT NULL operators instead.