International University, VNU-HCMC

School of Computer Science and Engineering

Lecture 5: SQL part 1

Instructor: Nguyen Thi Thuy Loan

nttloan@hcmiu.edu.vn, nthithuyloan@gmail.com
https://nttloan.wordpress.com/

Recap: Lecture 4

- Why use a DBMS?
- Structured data model: Relational data model
  - table, schema, instance, tuples, attributes
  - bag and set semantics
- Data independence
  - Physical independence: Can change how data is stored on disk without affecting applications
  - Logical independence: can change schema without affecting apps

Summary: Data Models

- The relational data model is the most standard for database management
  – and is the main focus of this course
- A semi-structured model/XML is also used in practice – you will use it in homework/ assignments.
- Unstructured data (text/photo/video) is unavoidable, but won't be covered in this class

Acknowledgement

- The following slides have been created by adapting the instructor material from the [RG] book, provided by the authors, Dr. Ramakrishnan and Dr. Gehrke.
- Other slides are referenced from Dr. Sudeepa Roy of Duke University.

International University, VNU-HCMC

Purpose of the Lecture

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Introduce the role of SQL in managing and querying relational databases.
- Present the basic structure of SQL (DDL, DML, DCL, TCL).
- Teach fundamental SQL commands (CREATE, SELECT, INSERT, UPDATE, DELETE).
- Demonstrate how SQL supports data retrieval and manipulation.

International University, VNU-HCMC

Warm-up Question

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Suppose you have a large dataset stored in a relational database (e.g., student records). How would you tell the computer to display specific information, such as only the students in class A?

International University, VNU-HCMC

Outlines

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Introduction
- Create database
- Create tables, constraints, and import data
  – Reading material: [RG] Chapters 3 and 5
  – Additional reading for practice: [GUW] Chapter 6

International University, VNU-HCMC

What is SQL?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

- SQL is the standard language for Relational Database System.

https://www.tutorialspoint.com

International University, VNU-HCMC

A Brief History of SQL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

1970 – Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.

1974 – Structured Query Language appeared.

1978 – IBM worked to develop Codd's ideas and released a product named System/R.

1986 – IBM developed the first prototype of a relational database and standardized it by ANSI. The first relational database was released by Relational Software, which later became known as Oracle.

International University, VNU-HCMC

A Brief History of SQL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Standards:
  - SQL-86
  - SQL-89 (minor revision)
  - SQL-92 (major revision)
  - SQL-99 (major extensions, current standard)
  - More: *MS SQL Server history* on the internet

Purposes of SQL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Data Manipulation Language (DML)
  - Querying: SELECT-FROM-WHERE
  - Modifying: INSERT/DELETE/UPDATE
- Data Definition Language (DDL)
  - CREATE/ALTER/DROP

DML - Data Manipulation Language

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Sr.No.	Command & Description
1	**SELECT** Retrieves certain records from one or more tables.
2	**INSERT** Creates a record.
3	**UPDATE** Modifies records.
4	**DELETE** Deletes records.

International University, VNU-HCMC

DDL - Data Definition Language

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Sr.No.	Command & Description
1	**CREATE** Creates a new table, a view of a table, or other object in the database.
2	**ALTER** Modifies an existing database object, such as a table.
3	**DROP** Deletes an entire table, a view of a table or other objects in the database.

International University, VNU-HCMC

Relational Model

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

column/ attribute/ field

row / tuple / record

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith1@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

- Mathematically, a relation is a set of tuples
  - Each tuple appears 0 or 1 times in the table
  - The order of the rows is unspecified

SQL (“sequel”)

- Standard query language for relational data
  - used for databases in many different contexts
  - inspires query languages for non-relational (e.g. SQL++)
- Everything not in quotes (‘…’) is case insensitive
- Provides standard types.

International University, VNU-HCMC

SQL (“sequel”)

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Examples:
  - numbers: INT, FLOAT, DECIMAL(p,s)
    - 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
  - strings: CHAR(n), VARCHAR(n)
    - CHAR(n): Fixed-length n
    - VARCHAR(n): Variable length. Maximum length n

SQL ("sequel") – Cont.

- BOOLEAN
- DATE, TIME, TIMESTAMP
  - DATE: Stores year, month, and day values
  - TIME: Stores hour, minute, and second values
  - TIMESTAMP: Stores year, month, day, hour, minute, and second values
- Additional types in here

Exact Numeric Data Types

DATA TYPE	FROM	TO
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
int	-2,147,483,648	2,147,483,647
smallint	-32,768	32,767
tinyint	0	255
bit	0	1
decimal	-10^38 +1	10^38 -1
numeric	-10^38 +1	10^38 -1
money	-922,337,203,685,477.5808	+922,337,203,685,477.5807
smallmoney	-214,748.3648	+214,748.3647

International University, VNU-HCMC

Approximate Numeric Data Types

DATA TYPE	FROM	TO
float	-1.79E + 308	1.79E + 308
real	-3.40E + 38	3.40E + 38

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

20

International University, VNU-HCMC

Date and Time Data Types

DATA TYPE	FROM	TO
datetime	Jan 1, 1753	Dec 31, 9999
smalldatetime	Jan 1, 1900	Jun 6, 2079
date	Stores a date like June 30, 1991	
time	Stores a time of day like 12:30 P.M.	

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

21

Character Strings Data Types

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Sr.No.	DATA TYPE & Description
1	**char** Maximum length of 8,000 characters.(Fixed length non-Unicode characters)
2	**varchar** Maximum of 8,000 characters.(Variable-length non-Unicode data).
3	**varchar(max)** Maximum length of 2E + 31 characters, Variable-length non-Unicode data (SQL Server 2005 only).
4	**text** Variable-length non-Unicode data with a maximum length of 2,147,483,647 characters.

International University, VNU-HCMC

Unicode Character Strings Data Types

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Sr.No.	DATA TYPE & Description
1	**nchar** Maximum length of 4,000 characters.(Fixed length Unicode)
2	**nvarchar** Maximum length of 4,000 characters.(Variable length Unicode)
3	**nvarchar(max)** Maximum length of 2E + 31 characters (SQL Server 2005 only).(Variable length Unicode)
4	**ntext** Maximum length of 1,073,741,823 characters. (Variable length Unicode)

International University, VNU-HCMC

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Binary Data Types

Sr.No.	DATA TYPE & Description
1	**binary** Maximum length of 8,000 bytes(Fixed-length binary data)
2	**varbinary** Maximum length of 8,000 bytes.(Variable length binary data)
3	**varbinary(max)** Maximum length of 2E + 31 bytes (SQL Server 2005 only). (Variable length Binary data)
4	**image** Maximum length of 2,147,483,647 bytes. (Variable length Binary Data)

International University, VNU-HCMC

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Miscellaneous Data Types

Sr.No.	DATA TYPE & Description
1	**sql_variant** Stores values of various SQL Server-supported data types, except text, ntext, and timestamp.
2	**timestamp** Stores a database-wide unique number that gets updated every time a row gets updated
3	**uniqueidentifier** Stores a globally unique identifier (GUID)
4	**xml** Stores XML data. You can store xml instances in a column or a variable (SQL Server 2005 only).

Demo on SQLite

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- E.g., type sqlite3
- https://www.db-book.com/university-lab-dir/sqljs.html

SQL statements

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Create database …
- create table …
- drop table ...
- alter table ... add/remove ...
- insert into ... values ...
- delete from ... where ...
- update ... set ... where ...

International University, VNU-HCMC

SQL statements

Syntax:

CREATE DATABASE DatabaseName:
CREATE DATABASE testDB;

DROP DATABASE DatabaseName:
DROP DATABASE testDB;

USE DatabaseName:
USE testDB;

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

International University, VNU-HCMC

CREATING TABLE

The basic syntax of the CREATE TABLE statement is as follows:

```
CREATE TABLE table_name(
        column1     datatype,
        Column2     datatype,
        column3     datatype,
        .....
        columnN     datatype,
        PRIMARY KEY (one or more columns)
);
```

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Creating Relation/Table

- Creates the "Students" relation
  - The type (domain) of each field is specified
  - enforced by the DBMS whenever tuples are added or modified
- As another example, the "Enrolled" table holds information about courses that students take

```
CREATE TABLE Students
    (sid CHAR(10),
    name CHAR(15),
    login CHAR(20),
    age INTEGER,
    gpa REAL/DECIMAL(2,1))

CREATE TABLE Enrolled
    (sid CHAR(20), ??
    cid CHAR(20),
    grade CHAR(2))
```

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Students

sid	cid	grade
53831	Carnatic101	C
53831	Reggae203	B
53650	Topology112	A
53666	History105	B

Enrolled

Example: CREATE TABLE

Customers(<u>ID</u>: int, name: string(20), age: int, address: string(25), salary decimal(18,2))

CREATE TABLE Customers(

);

Destroying Relation/table

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Syntax: Drop table Table_name

DROP TABLE Customers;

- Destroys the relation Customers
  – The schema information *and* the tuples are deleted.

Altering Relation/Table

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

```
ALTER TABLE      Students
    ADD COLUMN firstYear: integer
```

- The schema of Students is altered by adding a new field.
- What's the value in the new field?
  Every tuple in the current instance is extended with a *null* value in the new field.

Adding/ Insert into

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Syntax:

INSERT INTO TABLE_NAME (column1, column2, column3, ..., columnN) VALUES (value1, value2, value3,..., valueN);

INSERT INTO TABLE_NAME
VALUES (value1, value2, value3, ..., valueN);

Adding and Deleting Tuples

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Can insert a single tuple using:

  INSERT INTO Students (sid, name, login, age, gpa)
  VALUES (53688, 'Smith', 'smith@ee', 18, 3.2)

- Can delete all tuples satisfying some condition (e.g., name = Smith):

  DELETE
  FROM Students S
  WHERE S.name = 'Smith'

Example: Adding/ Insert into


```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```


update … set … where …

UPDATE Student
SET age = age + 2
where sid = '53680';

Integrity Constraints (ICs)

- IC: condition that must be true for any instance of the database
  - e.g., domain constraints
  - ICs are specified when the schema is defined
  - ICs are checked when relations are modified
- A legal instance of a relation satisfies all specified ICs
  - DBMS will not allow illegal instances
- If the DBMS checks ICs, stored data is more faithful to the real-world meaning
  - Avoids data entry errors, too!

Integrity Constraints

NOT NULL Constraint – Ensures that a column cannot have NULL value.

DEFAULT Constraint – Provides a default value for a column when none is specified.

UNIQUE Constraint – Ensures that all values in a column are different.

Integrity Constraints

PRIMARY Key – Uniquely identifies each row/record in a table.

FOREIGN Key – Uniquely identifies a row in any of the given table.

CHECK Constraint – Ensures that all the values in a column satisfies certain conditions.

INDEX – Used to create and retrieve data from the database very quickly.

Integrity Constraints-NOT NULL


```
CREATE TABLE CUSTOMERS(
        ID      INT             NOT NULL,
        NAME        VARCHAR (20)      NOT NULL,
        AGE     INT             NOT NULL,
        ADDRESS     CHAR (25) ,
        SALARY      DECIMAL (18, 2),
        PRIMARY KEY (ID)
);
```

If CUSTOMERS table has already been created
ALTER TABLE CUSTOMERS
ALTER COLUMN SALARY DECIMAL (18, 2) NOT NULL

International University, VNU-HCMC

Integrity Constraints-DEFAULT

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

```
CREATE TABLE CUSTOMERS(
        ID      INT             NOT NULL,
        NAME VARCHAR (20)       NOT NULL,
        AGE     INT             NOT NULL,
        ADDRESS      CHAR (25) ,
        SALARY       DECIMAL (18, 2) DEFAULT 5000.00,
        PRIMARY KEY (ID)
);
```

If the CUSTOMERS table has already been created

```
ALTER TABLE CUSTOMERS
        DROP column SALARY;

ALTER TABLE CUSTOMERS
        ADD SALARY DECIMAL (18, 2) DEFAULT 5000.00;
```

Duke CS, Fall 2024

43

International University, VNU-HCMC

Integrity Constraints-UNIQUE

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

```
CREATE TABLE CUSTOMERS(
        ID      INT             NOT NULL,
        NAME         VARCHAR (20)       NOT NULL
UNIQUE,
        AGE     INT             NOT NULL,
        ADDRESS      CHAR (25) ,
        SALARY       DECIMAL (18, 2),
        PRIMARY KEY (ID)
);
```

Duke CS, Fall 2024

44

Integrity Constraints-UNIQUE

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

If the CUSTOMERS table has already been created

```
ALTER TABLE CUSTOMERS
ADD CONSTRAINT UniqueConstraint UNIQUE(NAME);
```

DROP a UNIQUE Constraint

```
ALTER TABLE CUSTOMERS
        DROP CONSTRAINT UniqueConstraint;
```


International University, VNU-HCMC

Integrity Constraints-CHECK

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

```
CREATE TABLE CUSTOMERS(
        ID      INT     NOT NULL,
        NAME            VARCHAR (20)      NOT NULL,
        AGE     INT     NOT NULL CHECK (AGE >= 18),
        ADDRESS         CHAR (25) ,
        SALARY          DECIMAL (18, 2),
        PRIMARY KEY (ID)
);
```

Integrity Constraints-CHECK

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

If the CUSTOMERS table has already been created

```
ALTER TABLE CUSTOMERS
ADD CONSTRAINT CheckConstraint CHECK(AGE >=18);
```

DROP a CHECK Constraint

```
ALTER TABLE CUSTOMERS
        DROP CONSTRAINT CheckConstraint;
```


Integrity Constraints-INDEX

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Syntax:

```
CREATE INDEX index_name
        ON table_name ( column1, column2.....);
```

To create an INDEX on the AGE column, to optimize the search on customers for a specific age

```
CREATE INDEX idx_age
        ON CUSTOMERS ( AGE );
```

DROP an INDEX Constraint

```
ALTER TABLE CUSTOMERS
        DROP INDEX idx_age;
```

Review: Keys in a Database

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Key/ Candidate Key
- Primary Key
- Super Key
- Foreign Key

- Primary key attributes are <u>underlined</u> in a schema
  - Person(<u>pid</u>, address, name)
  - Person2(<u>address, name, age</u>, job)

Primary Key Constraints

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Key = subset of columns that uniquely identifies a tuple
- Another constraint on the table
  - No two tuples can have the same values for those columns
- Examples:
  - Movie(title, year, length, genre): key is (title, year)
  - What is a good key for a Student?

Primary Key Constraints

Students(sid: string, name: string, login: string, age: integer, gpa: real).

- Can have multiple keys for a table
- Only one of those keys may be "primary"
  - DBMS often makes searches by primary key faster
  - Other keys are called "secondary"

Primary and Candidate Keys in SQL

- Possibly many candidate keys
  - specified using UNIQUE
  - One of which is chosen as the primary key.

Example:

- "For a given student and course, there is a single grade."
- What is a primary key in a table?

```
CREATE TABLE Enrolled
  (sid CHAR(10)
   cid  CHAR(20),
   grade CHAR(2),
   PRIMARY KEY ???)
```

Primary and Candidate Keys in SQL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Possibly many candidate keys
  - specified using UNIQUE
  - one of which is chosen as the primary key.
- "For a given student and course, there is a single grade."

```
CREATE TABLE Enrolled
  (sid CHAR(10)
   cid  CHAR(20),
   grade CHAR(2),
   PRIMARY KEY     (sid,cid) )
```


Primary and Candidate Keys in SQL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Possibly many candidate keys
  - specified using UNIQUE
  - One of which is chosen as the primary key.
- "For a given student and course, there is a single grade." vs
- "Students can take only one course, and receive a single grade for that course; further, no two students in the same course receive the same grade."

```
CREATE TABLE Enrolled
( sid CHAR(10)
  cid  CHAR(20),
  grade CHAR(2),
  PRIMARY KEY (sid,cid))
```

```
CREATE TABLE Enrolled
(   sid CHAR(10)
   cid  CHAR(20),
   grade CHAR(2),
   PRIMARY KEY ???,
   UNIQUE ??? )
```

Primary and Candidate Keys in SQL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Possibly many candidate keys
  - specified using UNIQUE
  - One of which is chosen as the primary key.
- "For a given student and course, there is a single grade." vs
- "Students can take only one course, and receive a single grade for that course; further, no two students in the same course receive the same grade."

```
CREATE TABLE Enrolled
( sid CHAR(10)
  cid  CHAR(20),
  grade CHAR(2),
  PRIMARY KEY (sid,cid))
```

```
CREATE TABLE Enrolled
(   sid CHAR(10)
   cid  CHAR(20),
   grade CHAR(2),
   PRIMARY KEY sid,
   UNIQUE (cid, grade))
```


International University, VNU-HCMC

Primary and Candidate Keys in SQL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Possibly many candidate keys
  - specified using UNIQUE
  - One of which is chosen as the primary key.
- Used carelessly, an IC can prevent the storage of database instances that arise in practice!

```
CREATE TABLE Enrolled
( sid CHAR(10)
  cid  CHAR(20),
  grade CHAR(2),
  PRIMARY KEY (sid,cid))
```

```
CREATE TABLE Enrolled
(   sid CHAR(10)
   cid  CHAR(20),
   grade CHAR(2),
   PRIMARY KEY sid,
   UNIQUE (cid, grade))
```

Foreign Keys, Referential Integrity

- Foreign key: A Set of fields in one relation that is used to 'refer' to a tuple in another relation
  – Must correspond to the primary key of the second relation
  – Like a 'logical pointer'

- E.g., sid is a foreign key referring to Students:
  – Enrolled(sid: string, cid: string, grade: string)
  – If all foreign key constraints are enforced, referential integrity is achieved
  – i.e., no dangling references

Foreign Keys in SQL

- Only students listed in the Students relation should be allowed to enroll for courses

```
CREATE TABLE Enrolled
(sid CHAR(10), cid CHAR(20), grade CHAR(2),
    PRIMARY KEY    (sid,cid),
    FOREIGN KEY (sid) REFERENCES Students )
```

- Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

- Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Enforcing Foreign-Key Constraints

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

If there is a foreign-key constraint from relation *R* to relation S, two violations are possible:

1. An insert or update to *R* introduces values not found in *S*.
2. A deletion or update to S causes some tuples of *R* to "dangle."

International University, VNU-HCMC

Action taken

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Example: suppose R = Enrolled, S = Students

An insert or update to Enrolled that introduces a nonexistent Students must be rejected.

A delete or update to Students that removes a student value found in some tuples of Enrolled can be handled in four ways (next slide)

Action taken

1. *Default*: Reject the modification.

2. *Cascade*: Make the same changes in Enrolled.
   Deleted Students: delete Enrolled tuple.
   Updated Students: change value in Enrolled.

3. *Set NULL*: Change the Sid in E to NULL.
4. *Default is No action:* (delete/update is rejected)

Example: Cascade

Delete the 53666 tuple from Students:
Then delete all tuples from Enrolled that have sid = '53666'.

Update the 53666 tuple by changing '53666' to '53686':
Then change all Enrolled tuples with
sid = '53666' to sid = '53686'.

Example: Set NULL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Delete the 53666 tuple from Students:
Change all tuples of Enrolled that have
sid = '53666' to have sid = NULL.

Update the 53666 tuple by changing '53666' to '53686':
Same change as for deletion.

Referential Integrity in SQL

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- SQL/92 and SQL: 1999 support all 4 options on deletes and updates

```
CREATE TABLE Enrolled
( sid CHAR(10),
  cid CHAR(20),
  grade CHAR(2),
  PRIMARY KEY    (sid,cid),
  FOREIGN KEY (sid) REFERENCES
  Students
    ON DELETE CASCADE
    ON UPDATE SET DEFAULT )
```

Where do ICs Come From?

- ICs are based upon the semantics of the real-world enterprise that is being described in the database relations
- Key and foreign key ICs are the most common; more general ICs are supported, too.

Where do ICs Come From?

- Can we infer ICs from an instance?
  - We can check a database instance to see if an IC is violated, but we can NEVER infer that an IC is true by looking at an instance.
  - An IC is a statement about all possible instances!
  - For example, we know the name is not a key, but the assertion that sid is a key is given to us.

International University, VNU-HCMC

Example

- We want to store information about students and the courses they take.

Tables:

- Students(StudentID, Name, Major)
- Courses(CourseID, Title, Credits)
- Enrollments(StudentID, CourseID, Grade)

Please write SQL Commands to create tables (including primary keys and foreign keys) and insert data.

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

International University, VNU-HCMC

Example another Instances

- We will use these instances of the Sailors and Reserves relations in our examples
- If the key for the Reserves relation contained only the attributes *sid* and *bid*, how would the semantics differ?

Sailor

sid	sname	rating	age
22	dustin	7	45
31	lubber	8	55
58	rusty	10	35

Reserves

sid	bid	day
22	101	10/10/96
58	103	11/12/96

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Thank you for your attention!