**Dhaka University**

**Dept. of Computer Science and Engineering**

**Course Code: CSE 2201, Course Title: Database Systems**

**Database Systems Lab (Using Oracle Database 19c)**

* **Schemas of part of Banking Enterprise**

1. Independent Schemas:

*branch = (branch\_name, branch\_city, assets)*

*customer = (customer\_id, customer\_name, customer\_street, customer\_city, cell, dob)*

1. Dependent Schemas:

*account = (account\_no, branch\_name, balance)*

*loan = (loan\_no, branch\_name, amount)*

*depositor = (customer\_id, account\_no)*

*borrower =* (*customer\_id, loan\_no)*

* **DDL (Data Definition Language) (Command related to schema)**

1. create
2. describe
3. alter
   * add
   * modify
   * drop
4. rename
5. drop

* **Very common ‘data types’ used in Oracle (others available)**
* Character
* Numeric
* Date and time (date-time)
* Large Object (LOB)

1. **Character:** To store character (alphanumeric) data. Two types:
   1. char(size) – fixed length character literals (1 to 2000 bytes) [blank-padded to the fixed length]. Example: char(5)
   2. varchar2(size) - variable length character literals (1 to 4000 bytes). Example: varchar2(25)
2. **Numeric:** To store integers and real numbers in a **fixed-point** or **floating-point** format.

* The number datatype stores zero as well as positive and negative fixed numbers with absolute values from 1.0 x 10-130 to (but not including) 1.0 x 10126.
* Can be specified with a **precision** (p) and a **scale** (s) designator.
* Precision is the **total number of significant decimal digits** and scale is the number of digits from the **decimal point to the least significant digit**.
* You can specify a number datatype as follows:
  1. number(p) for an integer/whole number with precision p and scale 0, equivalent to number(p,0). Example: number(7)
  2. number(p,s) for a fixed-point number. Example: number(8,2)
  3. number for a floating-point number - The absence of precision and scale designators specifies the maximum range and precision for an Oracle number. Example: number

1. **Date and time:** default date format is DD-MON-RR. 22-MAR-25

* The RR enables you to store 20th century dates in the 21st century by specifying only the last two digits of the year.
* The valid date range is from January 1, 4712 BC to December 31, 9999 AD.
* Time is stored in a 24-hour format as HH24:MI:SS.
* Oracle Database supports the following datatypes:
  1. date [Example 09-DEC-11]
  2. timestamp [Example 09-DEC-11 02.05.49.000000 PM]

### 4. Storing Large Objects (LOB)

* Large Objects (LOBs) are a set of datatypes that are designed to hold large amounts of data.
* A LOB can hold up to a maximum size ranging from 8 terabytes to 128 terabytes depending on how your database is configured.
* The BLOB (Binary LOB) and CLOB (Character LOB) datatypes are internal LOB datatypes and are stored in the database.
  1. **Table Creation: (Using SQL Commands Window and Object Browser)**

Branch Table **(T10)**

create table branch

(branch\_name varchar2(15) primary key,

branch\_city varchar2(12) not null,

assets number(12) check (assets >= 100000));

Customer Table **(T20)**

create table customer

(customer\_id varchar2(5),

customer\_name varchar2(15),

customer\_street varchar2(12),

customer\_city varchar2(12) not null,

cell varchar2(11) unique,

dob date not null,

primary key (customer\_id));

Account Table **(One day)**

Where is **test**?

Loan Table

create table loan

(loan\_no char(5),

branch\_name varchar2(15),

amount number(10,2) not null,

primary key (loan\_no),

foreign key (branch\_name) references branch(branch\_name),

check (amount >= 0),

check (loan\_no like ‘L-%’));

Depositor Table

create table depositor

(customer\_id varchar2(5),

account\_no char(5),

primary key (customer\_id, account\_no),

foreign key (customer\_id) references customer(customer\_id),

foreign key (account\_no) references account(account\_no));

Borrower Table

create table borrower

(customer\_id varchar2(5),

loan\_no char(5),

primary key (customer\_id, loan\_no),

foreign key (customer\_id) references customer(customer\_id),

foreign key (loan\_no) references loan(loan\_no));

**2. Describing a Table:** Finding out the column details of a table created

Example: **describe** customer

**3. Modifying the structure of a table:**

*employee = (emp\_id, emp\_name, salary)*

* Adding new columns/keys:

Example: **alter table** employee

**add** (mobile **number**(11),

doj **date not null**);

**alter table** employee

**add primary key**(emp\_id)

* Modifying existing columns/data types/size:

Example: **alter table** employee

**modify** (emp\_name **varchar2**(35));

**alter table** employee

**modify** (mobile **varchar2**(11));

**alter table** employee

**rename** column emp\_name to employee\_name;

* Dropping columns/keys:

Example: **alter table** employee

**drop column** mobile;

**alter table** employee

**drop** primary key;

**4. Renaming Tables**

Example: **rename table** employee **to** emp

**5. Destroying Tables**

Example: **drop table** emp

* **DML (Data Manipulation Language)**

1. insert
2. update
3. delete
4. queries

* **Insertion of data into Tables:**

1. Single Record Insertion:

**insert into** branch

(branch\_name, branch\_city, assets)

**values** (‘Dhanmondi’, ‘Dhaka’, 10000000);

1. Multiple Records Insertion:

insert all

into branch values('Brighton', 'Brooklyn', 7100000)

into branch values ('Downtown', 'Brooklyn', 9000000)

into branch values ('Mianus', 'Horseneck', 400000)

into branch values ('Northtown', 'Rye', 3700000)

into branch values ('Perryridge','Horseneck', 1700000)

into branch values ('Pownal','Bennington', 300000)

into branch values ('Redwood', 'Palo Alto', 2100000)

into branch values ('Round Hill', 'Horseneck', 8000000)

select \* from dual

(is used to make the entire SQL INSERT ALL statement syntactically complete in **Oracle SQL**. In Oracle, when you use the INSERT ALL statement to insert multiple rows into a table, the syntax requires a final SELECT clause to execute the insert operation. The DUAL table is a special one-row, one-column table provided by Oracle specifically for these kinds of purposes—essentially a dummy table to allow expressions or operations to be executed without referencing actual data tables.

)

* **Queries**

SQL Clauses:

1. select
2. from
3. where
4. order by
5. group by
6. having

SQL Operations

1. Simple queries using ‘select’, ‘from’ and ‘where’ from one or more tables
2. Rename Operation
3. String Operation
4. Ordering
5. Set Operations (Union, Intersect, Minus)
6. Aggregate Functions
7. Null Values
8. Nested Sub-query (set-membership, set-comparison)
9. Views
10. Modification of the database (insert, update, delete)
11. Join Types and Conditions ( inner and outer)

* Commit and rollback