Project Description

Creating your dababase application with JDBC Develop a JAVA application based on the **examples (in /oracle/app/oracle/9.2.0.1.0/sqlj/demo)** that comes with Oracle. It will be sufficient to have a stand alone application that

- accommodates all the need of the intended users,
- allows the updating the database information, and
- provides an interface for query answering.

It is hoped that you come up with a system that is user friendly (menu driven, GUI etc.).

What is “accommodates all the need of the intended users”?

- Who are the intended users of your applications?
- What are the tasks that each type of users usually do (within the application)?

Your system should allow them to do their tasks!

Example

- The system: student information system
- Intended user groups: students, professors, cashiers, secretaries
- Students want to view their record
- The system should have a feature allowing a student to enter his id and obtains his transcript!

Example

- The system: student information system
- Intended user groups: students, professors, cashiers, secretaries
- Professors want to get the class listing
- The system should have a feature allowing a prof to enter his id and class id and obtains the class roster!

What is “allows the updating the database information”?

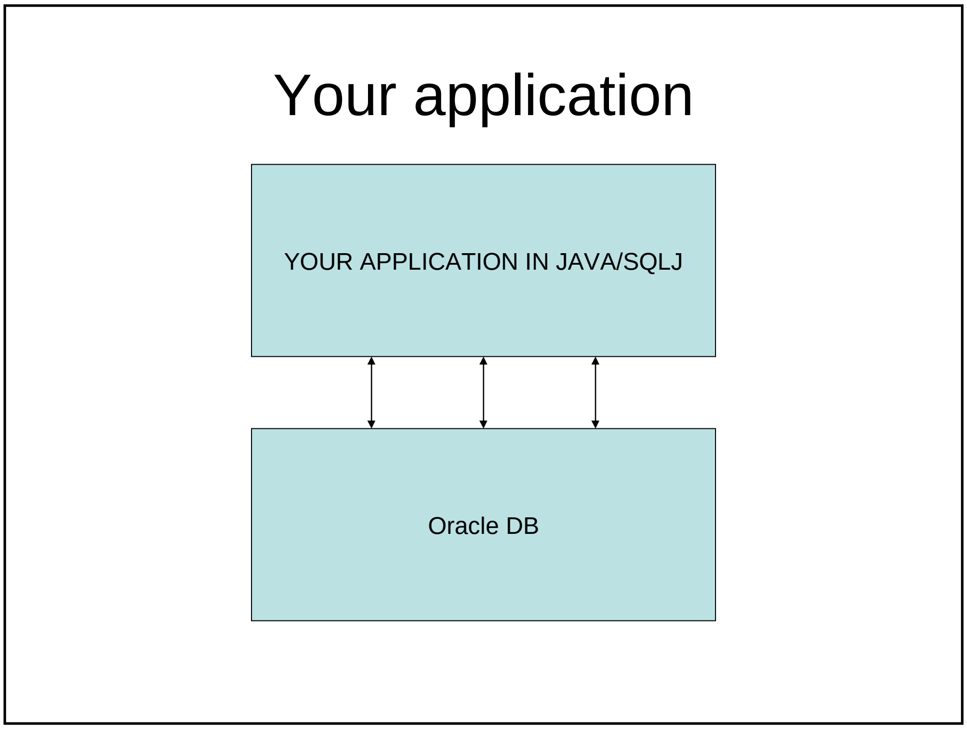
- Your system will be stored in Oracle – you will need to provide users a way to update information.
- Example: Student Information System
- We need to allow the registrar office to add new students, new courses, update students’ grades etc.

What is “provides an interface for query answering”?

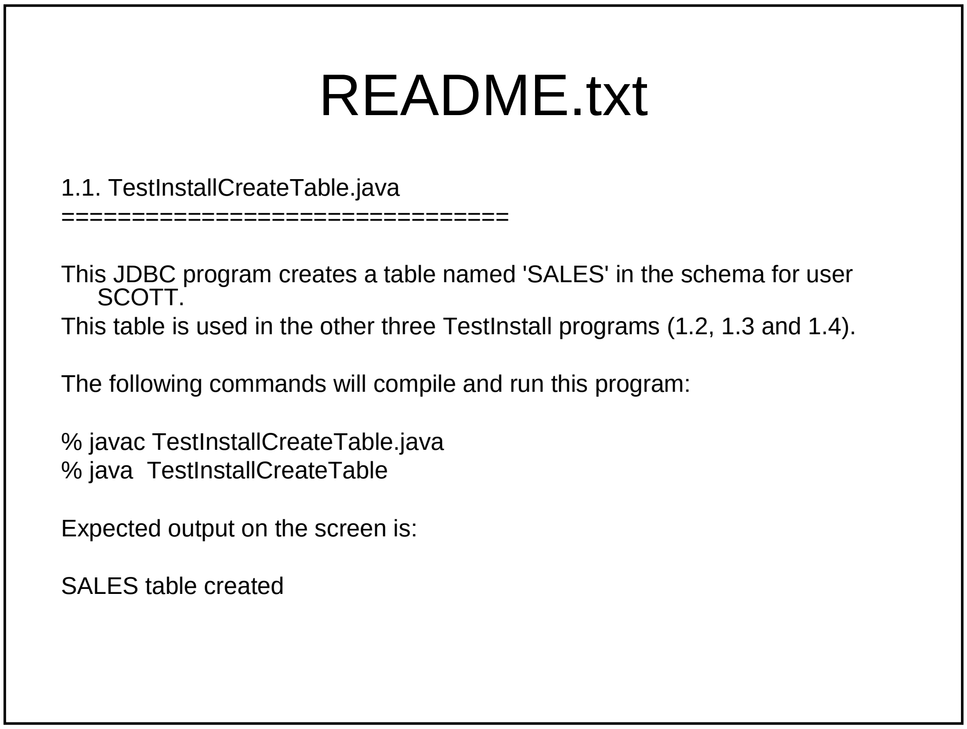
- Summary questions are often posted to a database application. Your application will not be an exception.
- You have thought about the ten most often asked questions in your application.
- You need to provide users who has a question in the ten frequently asked questions to ask your system one of the ten questions and get the answer from the system.

NOTE

- There is not always a clear distinction between “accommodates all the need of the intended users” and “provides an interface for query answering”. One might be used for another.



Your application



README.txt

1.1. TestInstallCreateTable.java
=================================

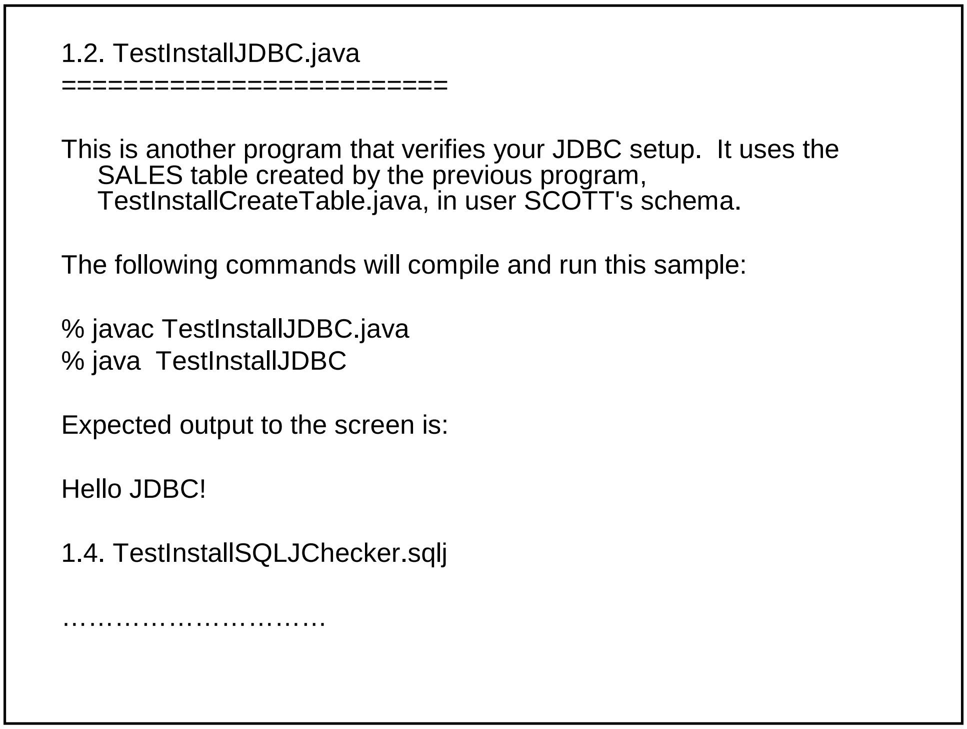
This JDBC program creates a table named 'SALES' in the schema for user SCOTT.
This table is used in the other three TestInstall programs (1.2, 1.3 and 1.4).

The following commands will compile and run this program:

```
% javac TestInstallCreateTable.java
% java  TestInstallCreateTable
```

Expected output on the screen is:

```
SALES table created
```



1.2. TestInstallJDBC.java
========================

This is another program that verifies your JDBC setup. It uses the SALES table created by the previous program, TestInstallCreateTable.java, in user SCOTT's schema.

The following commands will compile and run this sample:

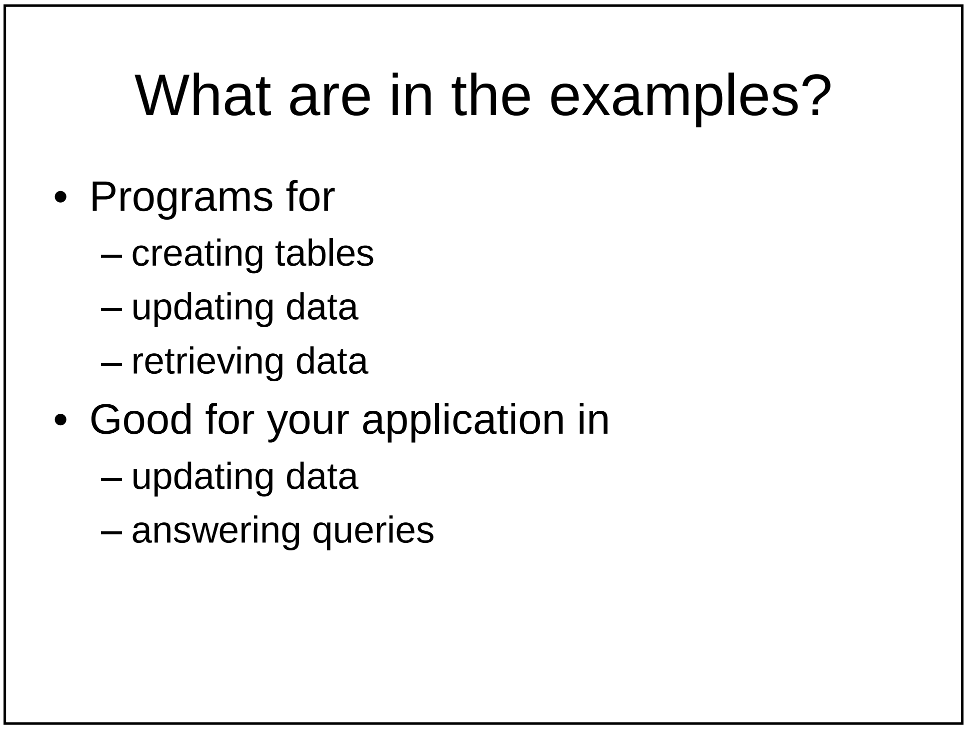
```
% javac TestInstallJDBC.java
% java  TestInstallJDBC
```

Expected output to the screen is:

```
Hello JDBC!
```

1.4. TestInstallSQLJChecker.sqlj

………………………..



What are in the examples?

- Programs for
  - creating tables
  - updating data
  - retrieving data
- Good for your application in
  - updating data
  - answering queries

What need to be done careful?

- The file “connect.properties” needs to be changed.
- Commit after each change you made

What is enough?

- A program that allows one to do all the required tasks:
  - Updating data (your application has tables t1, t2, t3, t4 – then the system will have to allow us to enter, modify, or delete data in all four tables)
  - Asking queries (display answer on screen)

Screen dump example (1)

application-team-XX

1. Updating
2. Querying
3. Exit

Select 1 – A new screen with 1,2,…,n tables appear for selection

Screen dump example (2)

application-team-XX

1. Student table
2. Professor table
3. Course table
4. Exit

Select 2 – display the professor table, ask whether the user want to enter new professor, edit some information, or delete

Screen dump example (3)

application-team-XX

1. Enter new
2. Modify old
3. Delete

Select: …..

And so it goes ….

Nicer – but not required!

- GUI interface
- Menu driven

- Applet?