Harmony IT Solution

Web Application Programming Interface (API)

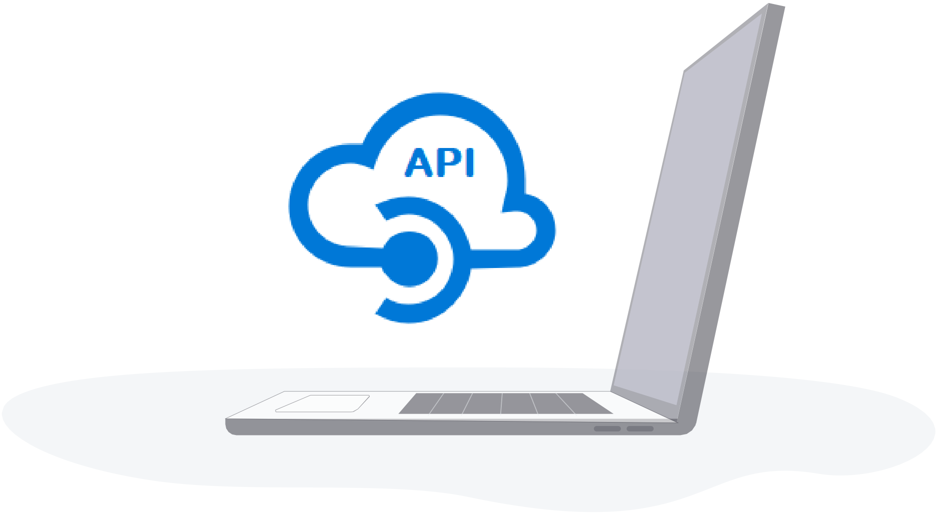
Tahaluf Training Center 2022

1 Create a Class Diagram

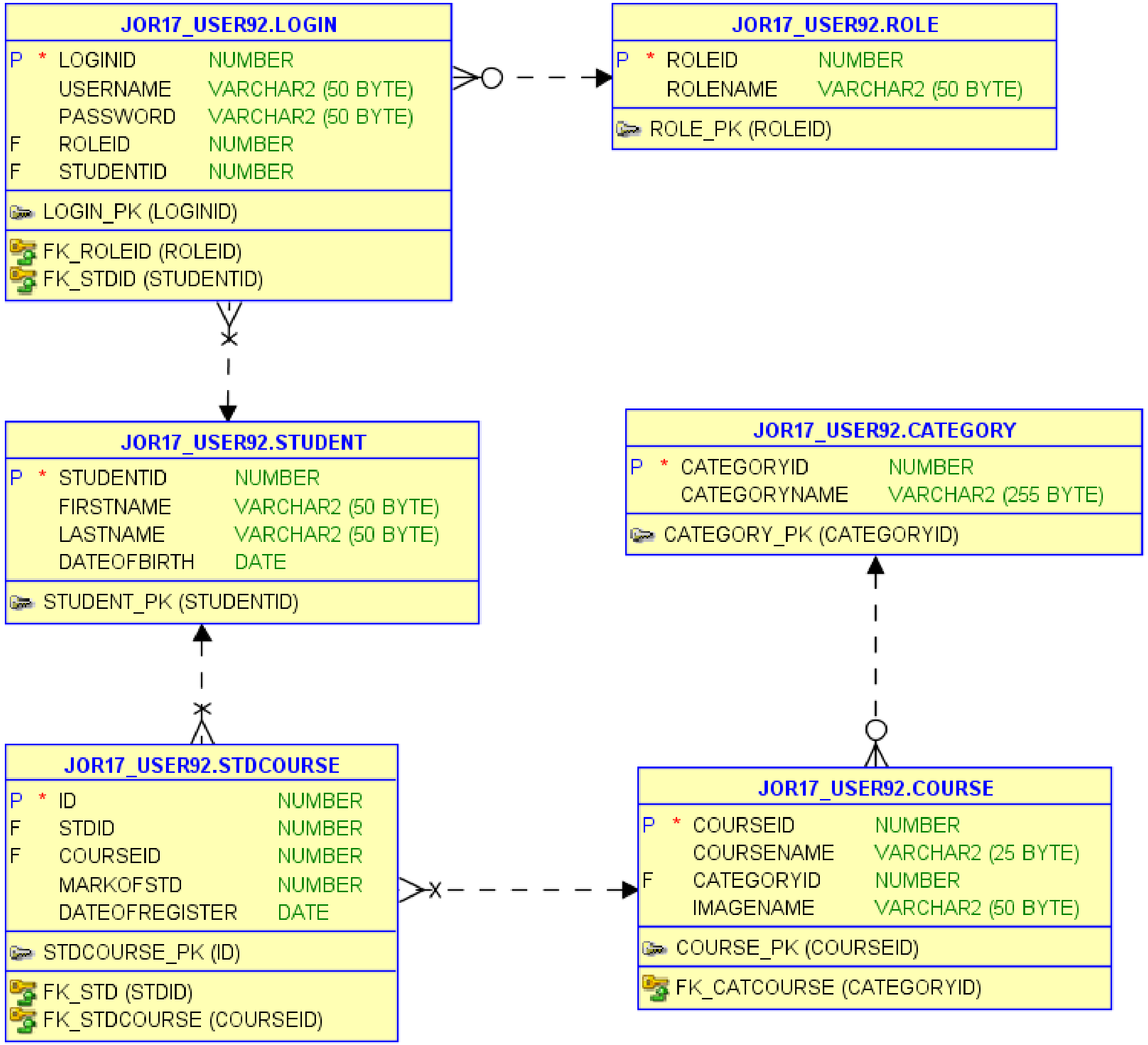
2 Overview of Package

3 Overview of Stored Procedure

4 Create Package And Stored Procedure



Create a Class Diagram



JOR17_USER92.LOGIN

P	*	LOGINID	NUMBER
		USERNAME	VARCHAR2 (50 BYTE)
		PASSWORD	VARCHAR2 (50 BYTE)
F		ROLEID	NUMBER
F		STUDENTID	NUMBER

LOGIN_PK (LOGINID)

FK_ROLEID (ROLEID)
FK_STDID (STUDENTID)

JOR17_USER92.ROLE

P	*	ROLEID	NUMBER
		ROLENAME	VARCHAR2 (50 BYTE)

ROLE_PK (ROLEID)

JOR17_USER92.STUDENT

P	*	STUDENTID	NUMBER
		FIRSTNAME	VARCHAR2 (50 BYTE)
		LASTNAME	VARCHAR2 (50 BYTE)
		DATEOFBIRTH	DATE

STUDENT_PK (STUDENTID)

JOR17_USER92.CATEGORY

P	*	CATEGORYID	NUMBER
		CATEGORYNAME	VARCHAR2 (255 BYTE)

CATEGORY_PK (CATEGORYID)

JOR17_USER92.STDCOURSE

P	*	ID	NUMBER
F		STDID	NUMBER
F		COURSEID	NUMBER
		MARKOFSTD	NUMBER
		DATEOFREGISTER	DATE

STDCOURSE_PK (ID)

FK_STD (STDID)
FK_STDCOURSE (COURSEID)

JOR17_USER92.COURSE

P	*	COURSEID	NUMBER
		COURSENAME	VARCHAR2 (25 BYTE)
F		CATEGORYID	NUMBER
		IMAGENAME	VARCHAR2 (50 BYTE)

COURSE_PK (COURSEID)

FK_CATCOURSE (CATEGORYID)

Overview of Package

A package is a schema object used to collect logically related PL/SQL variables, types and subprograms.

Packages have two parts, a specification (header) and a body.

The specification is the interface.

The body used to define the code for the subprograms and the queries for the cursors.

Overview of Stored Procedure

Stored procedures are similar to functions.

Stored procedure is created once and can be executed more than one time.

A stored procedure is created with a CREATE PROCEDURE statement and is executed with a CALL statement.

Create Package And Stored Procedure

Example 1

Create a course package that contains stored procedures to:

- display all courses in the database.
- Create a course.
- Update a course.
- Delete a course
- Get course by ID

Packages Specification

```
create or replace PACKAGE Course_Package
As
PROCEDURE GetAllCourses;
PROCEDURE GetCourseById(id in number) ;
PROCEDURE CREATECOURSE(COURSENAME IN course.coursename%TYPE, CATID
IN course.categoryid%TYPE, image in varchar);
PROCEDURE UPDATECOURSE( ID IN NUMBER ,CNAME IN
course.coursename%TYPE, CATID IN course.categoryid%TYPE, image in varchar);
PROCEDURE DeleteCourse(Id in number);
End Course_Package;
```

Packages Body

```
create or replace Package BODY Course_Package
As
PROCEDURE GetAllCourses
As
cur_all SYS_REFCURSOR ;
Begin
open cur_all for
Select * From course ;
Dbms_sql.return_result(cur_all);
End GetAllCourses ;
```

Packages Body

```
PROCEDURE GetCourseById(id in number)
As
Cur_item SYS_REFCURSOR;
Begin
open cur_item for
select * from course
where courseid = id;
Dbms_sql.return_result(cur_item);
End GetCourseById;
```

Packages Body

```
PROCEDURE CREATECOURSE(COURSENAME IN course.coursename%TYPE, CATID
IN course.categoryid%TYPE , image in varchar)
AS
id number ;
BEGIN
INSERT INTO COURSE VALUES (DEFAULT , COURSENAME , CATID , image );
COMMIT;

END CREATECOURSE;
```

Packages Body

```
PROCEDURE UPDATECOURSE( ID IN NUMBER ,CNAME IN
course.coursename%TYPE, CATID IN course.categoryid%TYPE , image in varchar)
AS
BEGIN
UPDATE COURSE
SET COURSENAME = CNAME , categoryid = CATID , imagename = image
WHERE COURSEID = ID ;
COMMIT;
END UPDATECOURSE;
```

Packages Body

```
PROCEDURE DeleteCourse(Id in number)
As
Begin
delete from course
where courseid = id ;
commit;
End DeleteCourse;

End Course_Package;
```

Example 2

Create a student package that contains stored procedures to:

- display all students in the database.
- Create a student.
- Update a student.
- Delete a student
- Get student by ID

Packages Specification

```
create or replace PACKAGE Student_Package AS
PROCEDURE GetAllStudent;
PROCEDURE CreateStudent(first_name IN VARCHAR,last_name in varchar,date_of_birth in date);
PROCEDURE UpdateStudent(ID IN NUMBER, first_name IN VARCHAR,last_name IN
VARCHAR,date_of_birth date);
PROCEDURE DeleteStudent(ID IN NUMBER);
PROCEDURE GetStudentById(ID IN NUMBER);
END Student_Package;
```

Packages Body

```
create or replace PACKAGE Body Student_Package as
PROCEDURE GetAllStudent
AS
c_all sys_refcursor;
BEGIN
open c_all for
select * from Student;
DBMS_SQL.RETURN_RESULT(c_all);
END GetAllStudent;
```

Packages Body

```
PROCEDURE CreateStudent(first_name IN VARCHAR,last_name in
varchar,date_of_birth in date)
IS
BEGIN
INSERT INTO Student (firstName ,lastname ,dateofbirth )
VALUES(first_name,last_name,date_of_birth);
COMMIT;
END CreateStudent;
```

Packages Body

```
PROCEDURE UpdateStudent(ID IN NUMBER, first_name IN VARCHAR,last_name
IN VARCHAR,date_of_birth date)
IS
BEGIN
Update Student SET firstname=first_name,lastname
=last_name,dateofbirth=date_of_birth
WHERE studentid =ID;
COMMIT;
END UpdateStudent;
```

Packages Body

```
PROCEDURE DeleteStudent(ID IN NUMBER)
IS
BEGIN
DELETE Student WHERE studentid =ID;
COMMIT;
END DeleteStudent;
```

Packages Body

```
PROCEDURE GetStudentById(ID IN NUMBER)
AS
c_all sys_refcursor;
BEGIN
OPEN c_all FOR
SELECT * FROM Student WHERE studentid =ID;
DBMS_SQL.RETURN_RESULT(c_all);
END GetStudentById;
END Student_Package;
```

Example 3

Create a studentCourse package that contains stored procedures to:

- display all studentCourse in the database.
- Create a studentCourse.
- Update a studentCourse.
- Delete a studentCourse
- Get studentCourse by ID

Packages Specification

```
create or replace PACKAGE stdcourse_Package AS
PROCEDURE GetAllStdCourse;
PROCEDURE CreateStdCourse(stdidid IN number,courseid in number,markof in number,
dateof_register in date);
PROCEDURE UpdateStdCourse(SCid in number, stdidid IN number,courseid in number,markof in
number,dateof_register in date);
PROCEDURE DeleteStdCourse(ID IN NUMBER);
PROCEDURE GetStdCourseById(ID IN NUMBER);
END stdcourse_Package;
```

Packages Body

```
create or replace PACKAGE Body stdcourse_Package as
PROCEDURE GetAllStdCourse
AS
c_all sys_refcursor;
BEGIN
open c_all for
select * from stdcourse;
DBMS_SQL.RETURN_RESULT(c_all);
END GetAllStdCourse;
```

Packages Body

```
PROCEDURE CreateStdCourse(stdidid IN number,courseid in number,markof in
number,dateof_register in date)
IS
BEGIN
INSERT INTO stdcourse (stdid ,courseid ,markofstd,dateofregister )
VALUES(stdidid,courseid,markof,dateof_register);
COMMIT;
END CreateStdCourse;
```

Packages Body

```
PROCEDURE UpdateStdCourse(SCid in number,stdidid IN number,courseid in
number,markof in number,dateof_register in date)
IS
BEGIN
Update stdcourse SET stdid = stdidid, courseid
=courseid,markofstd=markof,dateofregister=dateof_register
WHERE id =SCid;
COMMIT;
END UpdateStdCourse;
```

Packages Body

```
PROCEDURE DeleteStdCourse(ID IN NUMBER)
IS
BEGIN
DELETE stdcourse WHERE id =ID;
COMMIT;
END DeleteStdCourse;
```

Packages Body

```
PROCEDURE GetStdCourseById(ID IN NUMBER)
AS
c_all sys_refcursor;
BEGIN
OPEN c_all FOR
SELECT * FROM stdcourse WHERE courseid =ID;
DBMS_SQL.RETURN_RESULT(c_all);
END GetStdCourseById;
END stdcourse_Package;
```

Exercise

- ✓ Create a stored procedure to display FirstName and LastName from table student.
- ✓ Create a stored procedure to display student by firstName.
- ✓ Create a stored procedure to display student by BirthOfDate.
- ✓ Create a stored procedure to display a student by BirthOfDate interval.
- ✓ Create a stored procedure to display the students name with the highest n(3,4,…) marks

In Student_Packages Specification Add:

```
PROCEDURE GetStudentByFirstName(First_Name IN VARCHAR);
PROCEDURE GetStudentFNameAndLName;
PROCEDURE GetStudentByBirthdate(Birth_Date IN date);
PROCEDURE GetStudentBetweenInterval(DateFrom in date , DateTo in date);
procedure GetStudentsWithHighestMarks(NumOfStudent in number);
```

In Student_Packages Body Add:

```
PROCEDURE GetStudentByFirstName(First_Name IN VARCHAR)
AS
c_all sys_refcursor;
BEGIN
OPEN c_all for
SELECT * FROM Student WHERE FirstName=First_name;
DBMS_SQL.RETURN_RESULT(c_all);
END GetStudentByFirstName;
```

In Student_Packages Body Add:

```
PROCEDURE GetStudentFNameAndLName
AS
c_all sys_refcursor;
BEGIN
OPEN c_all FOR
SELECT FirstName,LastName FROM Student;
DBMS_SQL.RETURN_RESULT(c_all);
END GetStudentFNameAndLName;
```

In Student_Packages Body Add:

```
PROCEDURE GetStudentByBirthdate(Birth_Date IN date)
AS
c_all sys_refcursor;
BEGIN
OPEN c_all for
SELECT * FROM Student WHERE Trunc(DATEOFBIRTH) = Birth_Date;
DBMS_SQL.RETURN_RESULT(c_all);
END GetStudentByBirthdate;
```

In Student_Packages Body Add:

```
PROCEDURE GetStudentBetweenInterval(DateFrom in date , DateTo in date)
As
c_all SYS_REFCURSOR ;
Begin
open c_all for
select * from student
where dateofbirth >= datefrom and dateofbirth <= dateto;
dbms_sql.return_result(c_all);
End GetStudentBetweenInterval ;
```

In Student_Packages Body Add:

```
procedure GetStudentsWithHighestMarks(NumOfStudent in number)
As
c_all SYS_REFCURSOR;
Begin
open c_all for
select * from
(select s.*
from student s
inner join stdcourse sc
on s.studentid = sc.stdid
order by sc.markofstd desc)
where Rownum <= NumOfStudent;
Dbms_sql.return_result(c_all);
End GetStudentsWithHighestMarks;
```