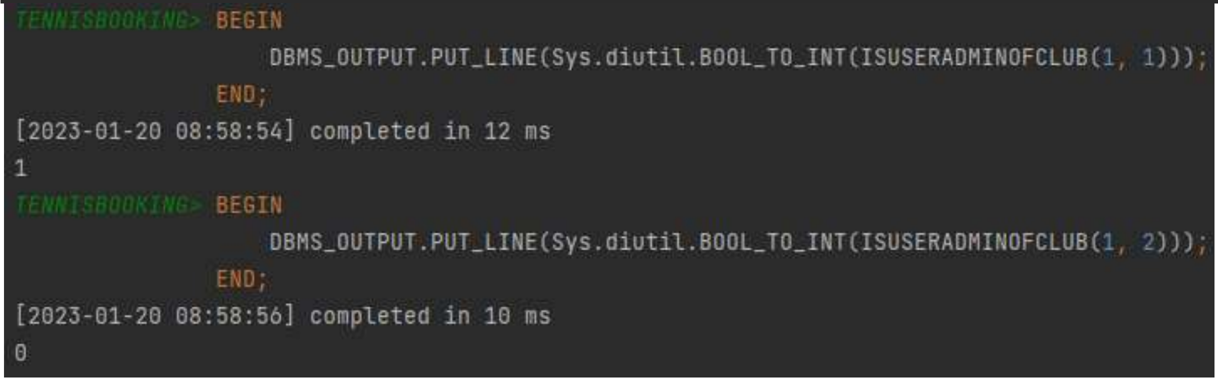
https://github.com/LuckyForce/DBI-PLSQL-Project-Winter-22-23

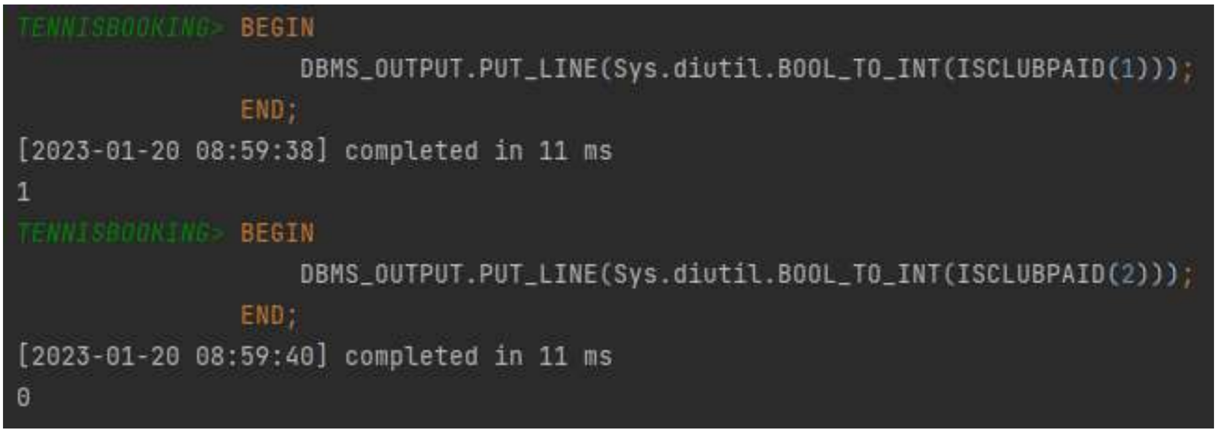
Functions

```
-- Check if Admin of Club
CREATE OR REPLACE FUNCTION isUserAdminOfClub(p_customer_id INTEGER,
p_club_id INTEGER)
RETURN BOOLEAN
AS
    v_is_admin BOOLEAN;
    v_club_id INTEGER;
    CURSOR v_clubs IS SELECT Id FROM Club WHERE AdminId = p_customer_id AND
Id = p_club_id;
BEGIN
    OPEN v_clubs;
    FETCH v_clubs INTO v_club_id;
    IF v_clubs%FOUND THEN
        v_is_admin := TRUE;
    ELSE
        v_is_admin := FALSE;
    END IF;
    CLOSE v_clubs;
    RETURN v_is_admin;
END;
```



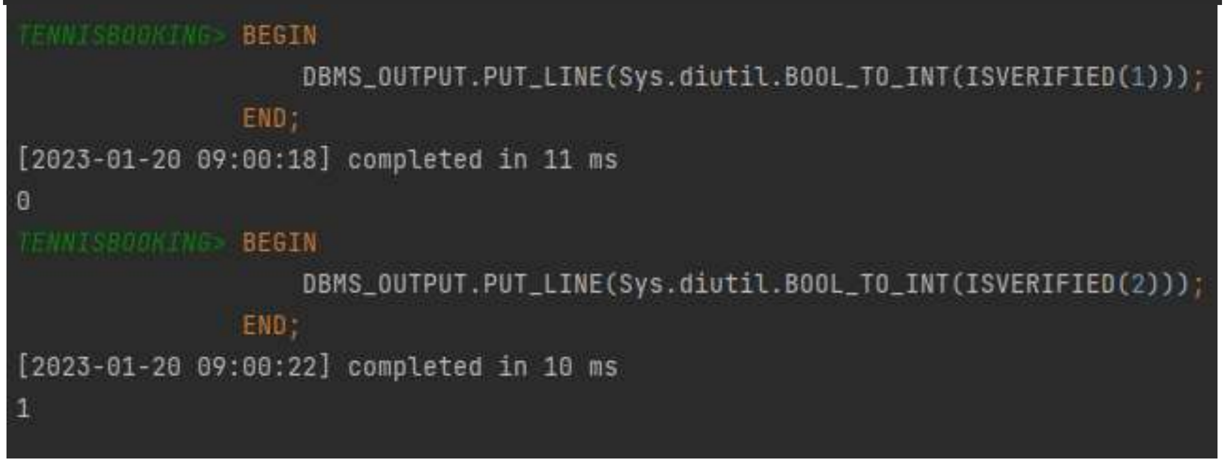
```
TENNISBOOKING> BEGIN
                   DBMS_OUTPUT.PUT_LINE(Sys.diutil.BOOL_TO_INT(ISUSERADMINOFCLUB(1, 1)));
               END;
[2023-01-20 08:58:54] completed in 12 ms
1
TENNISBOOKING> BEGIN
                   DBMS_OUTPUT.PUT_LINE(Sys.diutil.BOOL_TO_INT(ISUSERADMINOFCLUB(1, 2)));
               END;
[2023-01-20 08:58:56] completed in 10 ms
0
```

```
-- ClubPaid
CREATE OR REPLACE FUNCTION isClubPaid(p_club_id INTEGER)
RETURN BOOLEAN
AS
    v_is_paid BOOLEAN;
    v_paid_date DATE;
    CURSOR v_club IS SELECT PAIDTILL FROM Club WHERE Id = p_club_id;
BEGIN
    OPEN v_club;
    FETCH v_club INTO v_paid_date;
    IF v_club%NOTFOUND THEN
        v_is_paid := FALSE;
    ELSE
        IF v_paid_date > SYSDATE THEN
            v_is_paid := TRUE;
        ELSE
            v_is_paid := FALSE;
        END IF;
    END IF;
    CLOSE v_club;
    RETURN v_is_paid;
END;
```



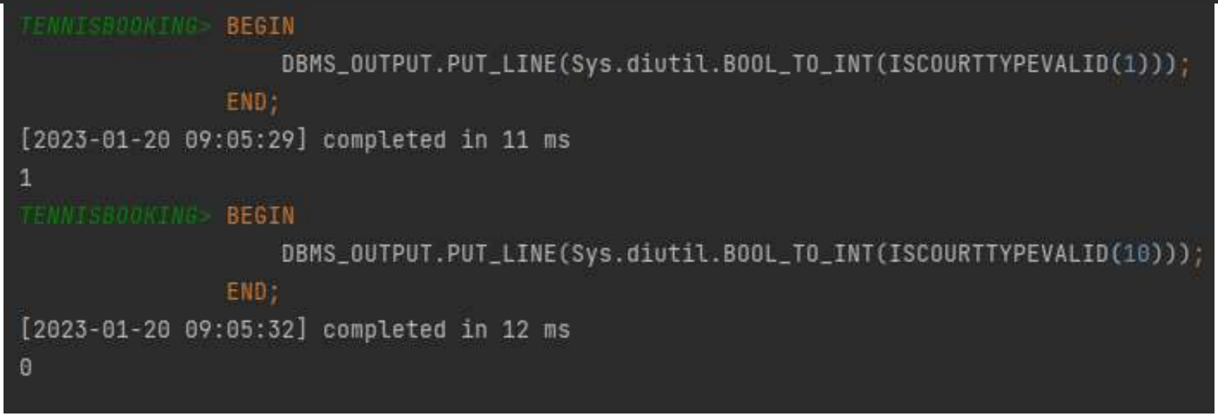
```
TENNISBOOKING> BEGIN
                   DBMS_OUTPUT.PUT_LINE(Sys.diutil.BOOL_TO_INT(ISCLUBPAID(1)));
               END;
[2023-01-20 08:59:38] completed in 11 ms
1
TENNISBOOKING> BEGIN
                   DBMS_OUTPUT.PUT_LINE(Sys.diutil.BOOL_TO_INT(ISCLUBPAID(2)));
               END;
[2023-01-20 08:59:40] completed in 11 ms
0
```

```
-- User Verified
CREATE OR REPLACE FUNCTION isVerified(p_customer_id INTEGER)
RETURN BOOLEAN
AS
    v_is_verified BOOLEAN;
    v_costumer_id INTEGER;
    CURSOR c_user IS SELECT ID FROM Customer WHERE Id = p_customer_id AND
Verified = 1;
BEGIN
    OPEN c_user;
    FETCH c_user INTO v_costumer_id;
    IF c_user%NOTFOUND THEN
        v_is_verified := FALSE;
    ELSE
        v_is_verified := TRUE;
    END IF;
    CLOSE c_user;
    RETURN v_is_verified;
END;
```



```
TENNISBOOKING> BEGIN
                   DBMS_OUTPUT.PUT_LINE(Sys.diutil.BOOL_TO_INT(ISVERIFIED(1)));
               END;
[2023-01-20 09:00:18] completed in 11 ms
0
TENNISBOOKING> BEGIN
                   DBMS_OUTPUT.PUT_LINE(Sys.diutil.BOOL_TO_INT(ISVERIFIED(2)));
               END;
[2023-01-20 09:00:22] completed in 10 ms
1
```

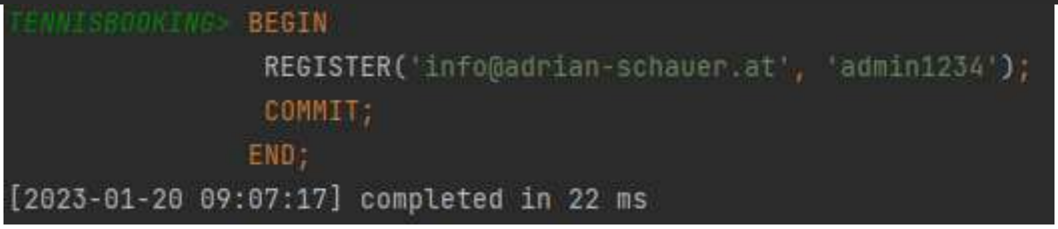
```
-- Court Type Valid
CREATE OR REPLACE FUNCTION isCourtTypeValid(p_court_type_id INTEGER)
RETURN BOOLEAN
AS
    v_is_valid BOOLEAN;
    v_court_type_id INTEGER;
    CURSOR v_court_type IS SELECT ID FROM CourtType WHERE Id =
p_court_type_id;
BEGIN
    OPEN v_court_type;
    FETCH v_court_type INTO v_court_type_id;
    IF v_court_type%NOTFOUND THEN
        v_is_valid := FALSE;
    ELSE
        v_is_valid := TRUE;
    END IF;
    CLOSE v_court_type;
    RETURN v_is_valid;
END;
```



```
TENNISBOOKING> BEGIN
                   DBMS_OUTPUT.PUT_LINE(Sys.diutil.BOOL_TO_INT(ISCOURTTYPEVALID(1)));
               END;
[2023-01-20 09:05:29] completed in 11 ms
1
TENNISBOOKING> BEGIN
                   DBMS_OUTPUT.PUT_LINE(Sys.diutil.BOOL_TO_INT(ISCOURTTYPEVALID(10)));
               END;
[2023-01-20 09:05:32] completed in 12 ms
0
```

Procedures

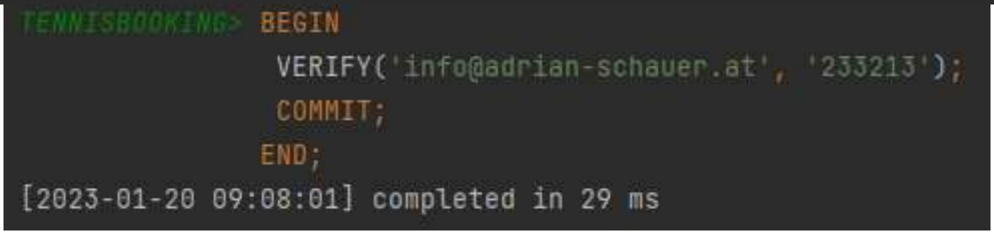
```
-- Procedure register(email, password)
CREATE OR REPLACE PROCEDURE register(p_email IN VARCHAR2, p_password IN
VARCHAR2)
IS
    v_user_id NUMBER;
    CURSOR c_customer IS
        SELECT ID
        FROM customer
        WHERE email = p_email;
BEGIN
    OPEN c_customer;
    FETCH c_customer INTO v_user_id;
    IF c_customer%FOUND THEN
        CLOSE c_customer;
        RAISE_APPLICATION_ERROR(-20001, 'Email already exists');
    END IF;
    CLOSE c_customer;
    INSERT INTO customer (uuid, email, password, verificationcode,
verified, resetcode, firstname, lastname, gender, welcomed)
    VALUES (sys_guid(), p_email, p_password, '233213', 0, '132321', 'Some
Name', 'Some Name', 1, 0);
END;
```



```
TENNISBOOKING> BEGIN
                   REGISTER('info@adrian-schauer.at', 'admin1234');
                   COMMIT;
               END;
[2023-01-20 09:07:17] completed in 22 ms
```

```
12    12 [illegible] info@adrian-schauer.at  admin1234  233213      0 132321  Some Name  Some Name  1
```

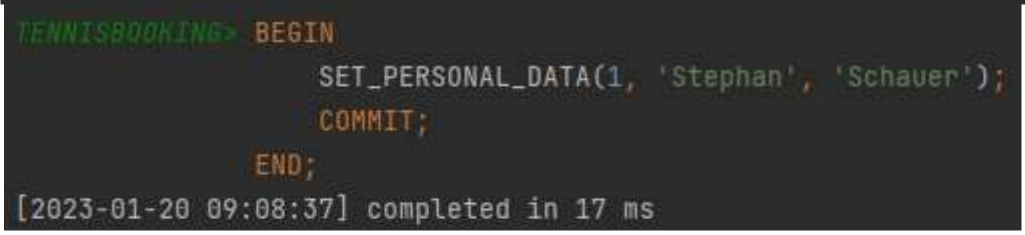
```
-- Procedure verify(email, verificationcode)
CREATE OR REPLACE PROCEDURE verify(p_email IN VARCHAR2, p_verificationcode
IN VARCHAR2)
IS
    v_user_id NUMBER;
    CURSOR c_customer IS
        SELECT ID
        FROM customer
        WHERE email = p_email AND verificationcode = p_verificationcode;
BEGIN
    OPEN c_customer;
    FETCH c_customer INTO v_user_id;
    IF NOT c_customer%FOUND THEN
        CLOSE c_customer;
        RAISE_APPLICATION_ERROR(-20002, 'Customer does not exist');
    END IF;
    CLOSE c_customer;
    UPDATE customer
    SET verified = 1
    WHERE email = p_email AND verificationcode = p_verificationcode;
END;
```



```
TENNISBOOKING> BEGIN
                VERIFY('info@adrian-schauer.at', '233213');
                COMMIT;
               END;
[2023-01-20 09:08:01] completed in 29 ms
```

```
12    12 [illegible]  info@adrian-schauer.at  admin123A  233213    1 132321    Some Name    Some Name    1
```

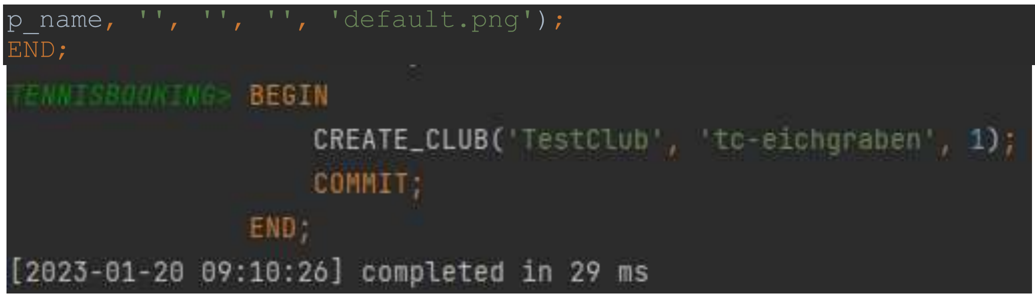
```
-- Procedure set_personal_data(FirstName, LastName, Gender, BirthDate,
user_id)
CREATE OR REPLACE PROCEDURE set_personal_data(p_costumer_id NUMBER,
p_firstname IN VARCHAR2, p_lastname IN VARCHAR2)
IS
    v_verified BOOLEAN := isverified(p_costumer_id);
BEGIN
    IF v_verified THEN
        RAISE_APPLICATION_ERROR(-20003, 'Customer is not verified');
    END IF;

    UPDATE customer
    SET firstname = p_firstname, lastname = p_lastname
    WHERE id = p_costumer_id;
END;
```



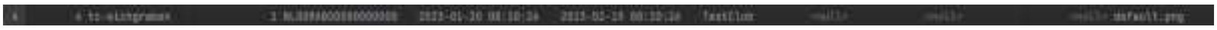
```
TENNISBOOKING> BEGIN
                SET_PERSONAL_DATA(1, 'Stephan', 'Schauer');
                COMMIT;
               END;
[2023-01-20 09:08:37] completed in 17 ms
```

```
1    1 [illegible]    info@adrian-schauer.at 123456    [illegible]    0 [illegible]    Stephan    Schauer    0
```

```
-- Procedure create_club(name, link, user)
CREATE OR REPLACE PROCEDURE create_club(p_name IN VARCHAR2, p_link IN
VARCHAR2, p_user_id NUMBER)
IS
    v_verified BOOLEAN := isverified(p_user_id);
BEGIN
    IF v_verified THEN
        RAISE_APPLICATION_ERROR(-20003, 'Customer is not verified');
    END IF;

    INSERT INTO CLUB (LINK, ADMINID, IBAN, PAIDTILL, FREETRIALTILL, NAME,
INFO, ADDRESS, ZIPCODE, IMAGEPATH)
    VALUES (p_link, p_user_id, 'NL00RABO0000000000', SYSDATE, SYSDATE + 30,
p_name, '', '', '', 'default.png');
END;
```

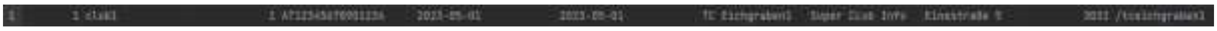


```
TENNISBOOKING> BEGIN
                   CREATE_CLUB('TestClub', 'tc-eichgraben', 1);
                   COMMIT;
               END;
[2023-01-20 09:10:26] completed in 29 ms
```

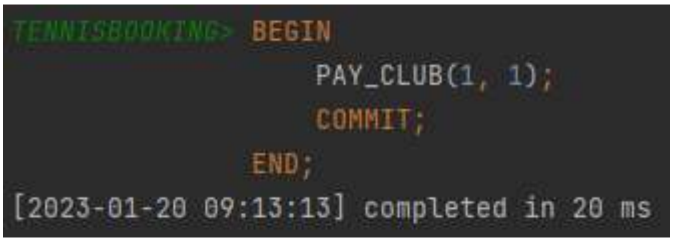


```
4    4 tc-eichgraben    1 NL0000000000000000    2023-01-20 08:10:26    2023-02-19 08:10:26    TestClub    <null>    <null>    <null> default.png
```

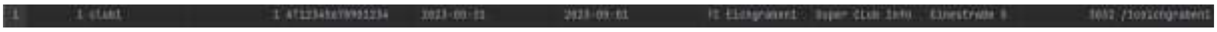
```
-- Procedure pay_club(club_id, user_id)
CREATE OR REPLACE PROCEDURE pay_club(p_club_id NUMBER, p_user_id NUMBER)
IS
    v_verified BOOLEAN := isverified(p_user_id);
    v_is_admin BOOLEAN := ISUSERADMINOFCLUB(p_club_id, p_user_id);
BEGIN
    IF v_verified THEN
        RAISE_APPLICATION_ERROR(-20003, 'Customer is not verified');
    END IF;
    IF NOT v_is_admin THEN
        RAISE_APPLICATION_ERROR(-20004, 'User is not admin of club');
    END IF;

    UPDATE CLUB
    SET PAIDTILL = PAIDTILL + 30
    WHERE ID = p_club_id;
END;
```



```
1    1 club1    1 AT12345678901234    2023-05-01    2023-05-01    TC Eichgraben1    Super Club Info    Kinostraße 5    3032 /tceichgraben1
```

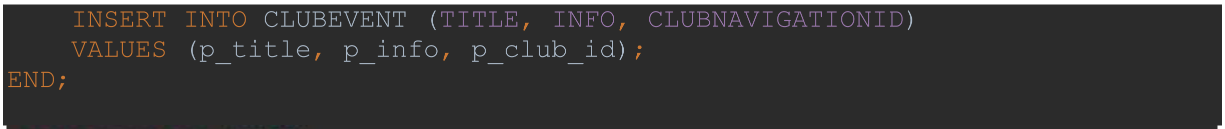


```
TENNISBOOKING> BEGIN
                   PAY_CLUB(1, 1);
                   COMMIT;
               END;
[2023-01-20 09:13:13] completed in 20 ms
```

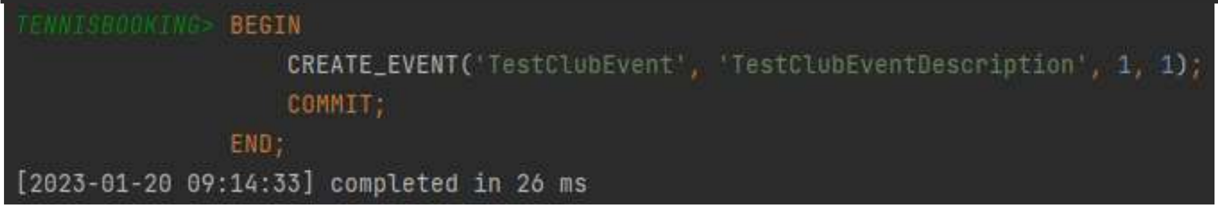


```
1    1 club1    1 AT12345678901234    2023-05-31    2023-05-01    TC Eichgraben1    Super Club Info    Kinostraße 5    3032 /tceichgraben1
```

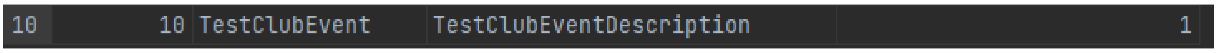
```
-- Procedure create_event(name, description, date, time, location, club_id,
user_id)
CREATE OR REPLACE PROCEDURE create_event(p_title IN VARCHAR2, p_info IN
VARCHAR2, p_club_id NUMBER, p_user_id NUMBER)
IS
    v_verified BOOLEAN := isverified(p_user_id);
    v_is_admin BOOLEAN := ISUSERADMINOFCLUB(p_user_id, p_club_id);
    v_is_paid BOOLEAN := ISCLUBPAID(p_club_id);
BEGIN
    IF v_verified THEN
        RAISE_APPLICATION_ERROR(-20003, 'Customer is not verified');
    END IF;

    IF NOT v_is_admin THEN
        RAISE_APPLICATION_ERROR(-20004, 'Customer is not admin of club');
    END IF;

    IF NOT v_is_paid THEN
        RAISE_APPLICATION_ERROR(-20005, 'Club is not paid');
    END IF;
```



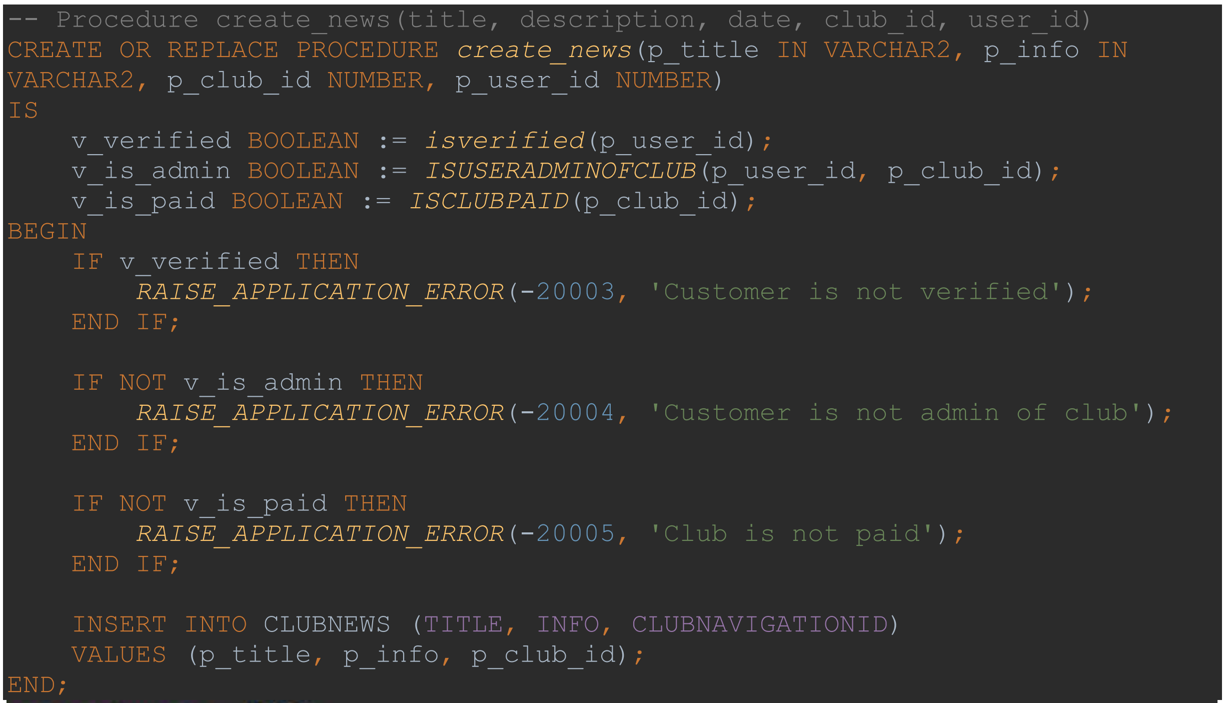
```
    INSERT INTO CLUBEVENT (TITLE, INFO, CLUBNAVIGATIONID)
    VALUES (p_title, p_info, p_club_id);
END;
```



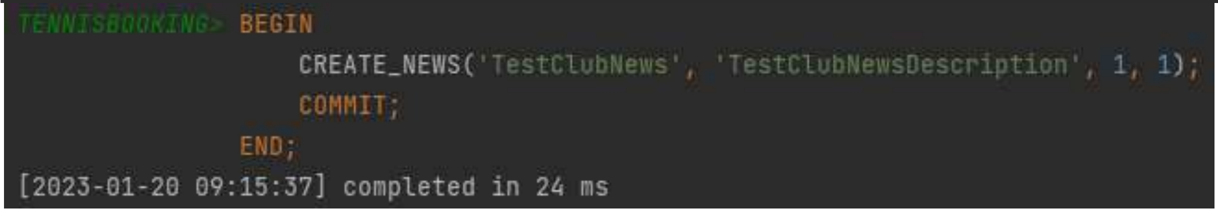
```
TENNISBOOKING> BEGIN
                   CREATE_EVENT('TestClubEvent', 'TestClubEventDescription', 1, 1);
                   COMMIT;
               END;
[2023-01-20 09:14:33] completed in 26 ms
```



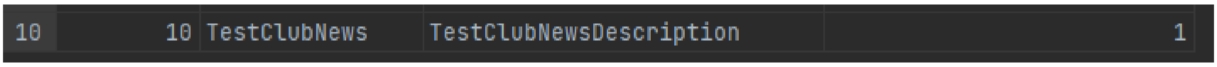
10	10	TestClubEvent	TestClubEventDescription		1



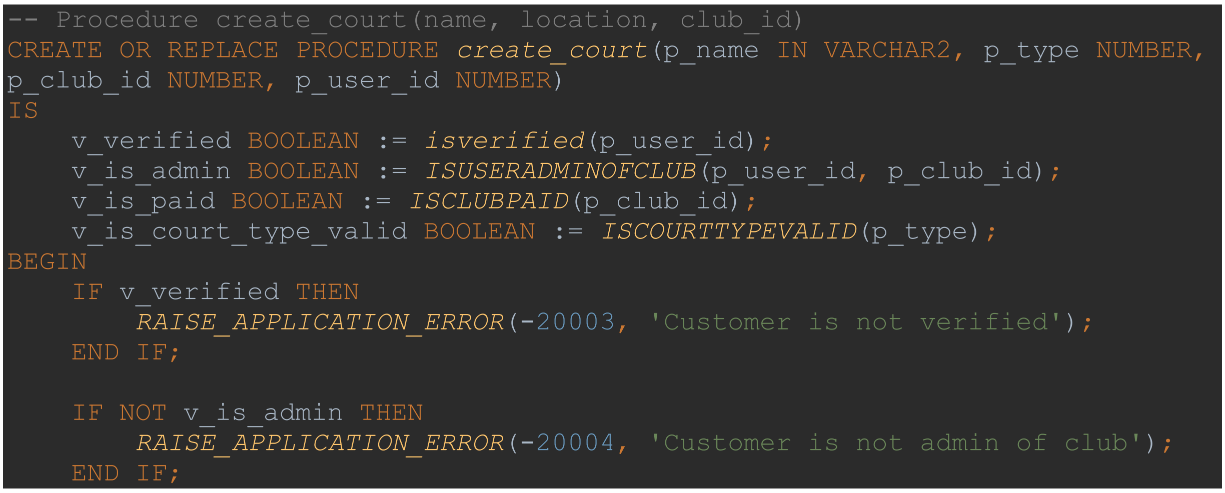
```
-- Procedure create_news(title, description, date, club_id, user_id)
CREATE OR REPLACE PROCEDURE create_news(p_title IN VARCHAR2, p_info IN
VARCHAR2, p_club_id NUMBER, p_user_id NUMBER)
IS
    v_verified BOOLEAN := isverified(p_user_id);
    v_is_admin BOOLEAN := ISUSERADMINOFCLUB(p_user_id, p_club_id);
    v_is_paid BOOLEAN := ISCLUBPAID(p_club_id);
BEGIN
    IF v_verified THEN
        RAISE_APPLICATION_ERROR(-20003, 'Customer is not verified');
    END IF;

    IF NOT v_is_admin THEN
        RAISE_APPLICATION_ERROR(-20004, 'Customer is not admin of club');
    END IF;

    IF NOT v_is_paid THEN
        RAISE_APPLICATION_ERROR(-20005, 'Club is not paid');
    END IF;

    INSERT INTO CLUBNEWS (TITLE, INFO, CLUBNAVIGATIONID)
    VALUES (p_title, p_info, p_club_id);
END;
```



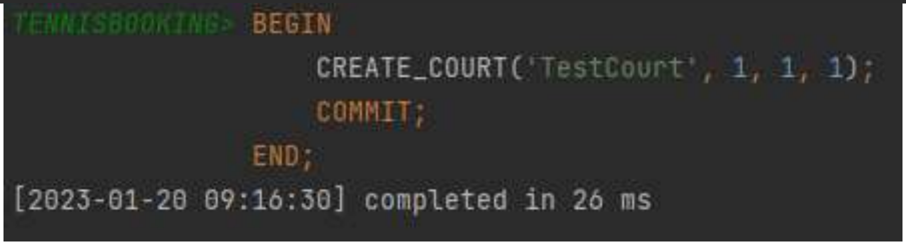
```
TENNISBOOKING> BEGIN
                   CREATE_NEWS('TestClubNews', 'TestClubNewsDescription', 1, 1);
                   COMMIT;
               END;
[2023-01-20 09:15:37] completed in 24 ms
```



10	10	TestClubNews	TestClubNewsDescription		1



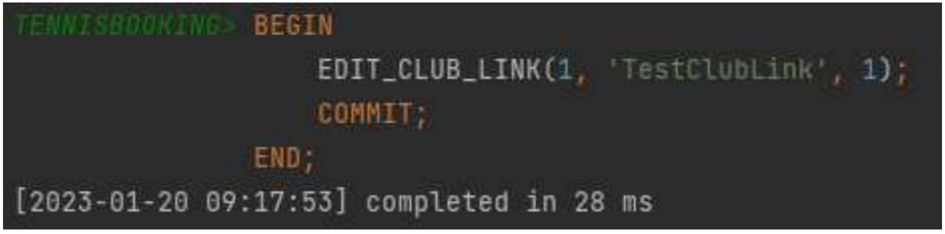
```
-- Procedure create_court(name, location, club_id)
CREATE OR REPLACE PROCEDURE create_court(p_name IN VARCHAR2, p_type NUMBER,
p_club_id NUMBER, p_user_id NUMBER)
IS
    v_verified BOOLEAN := isverified(p_user_id);
    v_is_admin BOOLEAN := ISUSERADMINOFCLUB(p_user_id, p_club_id);
    v_is_paid BOOLEAN := ISCLUBPAID(p_club_id);
    v_is_court_type_valid BOOLEAN := ISCOURTTYPEVALID(p_type);
BEGIN
    IF v_verified THEN
        RAISE_APPLICATION_ERROR(-20003, 'Customer is not verified');
    END IF;

    IF NOT v_is_admin THEN
        RAISE_APPLICATION_ERROR(-20004, 'Customer is not admin of club');
    END IF;
    IF NOT v_is_paid THEN
        RAISE_APPLICATION_ERROR(-20005, 'Club is not paid');
    END IF;

    IF NOT v_is_court_type_valid THEN
        RAISE_APPLICATION_ERROR(-20006, 'Court type is not valid');
    END IF;

    INSERT INTO COURT (NAME, BOOKABLE, TYPE, APRICE, BPRICE, ATIMEFROM,
ATIMETILL, AWEEKENDTIMETILL, CLUBNAVIGATIONID)
    VALUES (p_name, 1, 1, 10, 10, 10, 10, 10, p_club_id);
END;
```



```
TENNISBOOKING> BEGIN
                   CREATE_COURT('TestCourt', 1, 1, 1);
                   COMMIT;
               END;
[2023-01-20 09:16:30] completed in 26 ms
```

```
10    10 TestCourt    1    1 10    10    10    10    10    1
```

```
-- Procedure edit_club_link(club_id, link)
CREATE OR REPLACE PROCEDURE edit_club_link(p_club_id NUMBER, p_link IN
VARCHAR2, p_user_id NUMBER)
IS
    v_verified BOOLEAN := isverified(p_user_id);
    v_is_admin BOOLEAN := ISUSERADMINOFCLUB(p_user_id, p_club_id);
    v_is_paid BOOLEAN := ISCLUBPAID(p_club_id);
BEGIN
    IF v_verified THEN
        RAISE_APPLICATION_ERROR(-20003, 'Customer is not verified');
    END IF;

    IF NOT v_is_admin THEN
        RAISE_APPLICATION_ERROR(-20004, 'Customer is not admin of club');
    END IF;

    IF NOT v_is_paid THEN
        RAISE_APPLICATION_ERROR(-20005, 'Club is not paid');
    END IF;

    UPDATE CLUB
    SET LINK = p_link
    WHERE ID = p_club_id;
END;
```

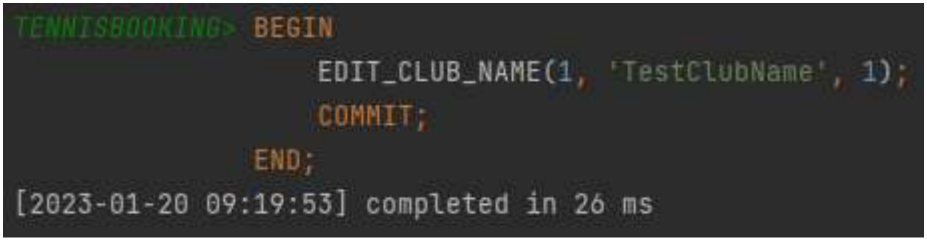


```
TENNISBOOKING> BEGIN
                   EDIT_CLUB_LINK(1, 'TestClubLink', 1);
                   COMMIT;
               END;
[2023-01-20 09:17:53] completed in 28 ms
```

```
-- Procedure edit_club_name(club_id, name)
CREATE OR REPLACE PROCEDURE edit_club_name(p_club_id NUMBER, p_name IN
VARCHAR2, p_user_id NUMBER)
IS
```

```
    v_verified BOOLEAN := isverified(p_user_id);
    v_is_admin BOOLEAN := ISUSERADMINOFCLUB(p_user_id, p_club_id);
    v_is_paid BOOLEAN := ISCLUBPAID(p_club_id);
BEGIN
    IF v_verified THEN
        RAISE_APPLICATION_ERROR(-20003, 'Customer is not verified');
    END IF;

    IF NOT v_is_admin THEN
        RAISE_APPLICATION_ERROR(-20004, 'Customer is not admin of club');
    END IF;

    IF NOT v_is_paid THEN
        RAISE_APPLICATION_ERROR(-20005, 'Club is not paid');
    END IF;

    UPDATE CLUB
    SET NAME = p_name
    WHERE ID = p_club_id;
END;
```

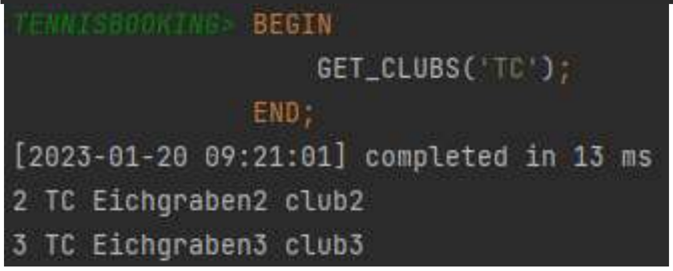
```
1    1 TestClubLink    1 AT12345678901234    2023-05-31    2023-05-01    TC Eichgraben1    Super Club Info    Einestraße 5    3032 /tceichgraben1
```



```
TENNISBOOKING> BEGIN
                   EDIT_CLUB_NAME(1, 'TestClubName', 1);
                   COMMIT;
               END;
[2023-01-20 09:19:53] completed in 26 ms
```

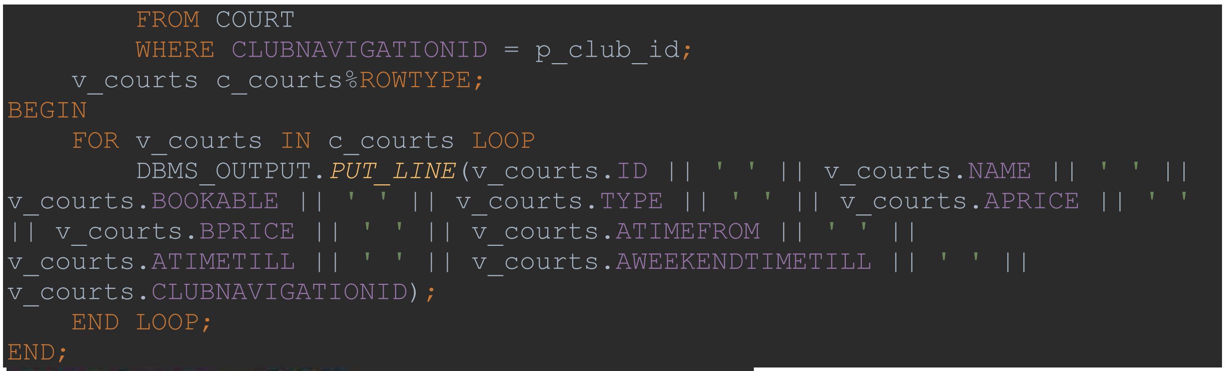
```
1    1 TestClubLink    1 AT12345678901234    2023-05-31    2023-05-01    TestClubName    Super Club Info    Einestraße 5    3032 /tceichgraben1
```

```
-- Get all clubs
CREATE OR REPLACE PROCEDURE get_clubs(p_search IN VARCHAR2)
IS
    CURSOR c_clubs IS
        SELECT ID, NAME, LINK
        FROM CLUB
        WHERE NAME LIKE '%' || p_search || '%';
    v_clubs c_clubs%ROWTYPE;
BEGIN
    IF p_search IS NULL THEN
        RAISE_APPLICATION_ERROR(-20001, 'Search is null');
    END IF;
    FOR v_clubs IN c_clubs LOOP
        DBMS_OUTPUT.PUT_LINE(v_clubs.ID || ' ' || v_clubs.NAME || ' ' ||
v_clubs.LINK);
    END LOOP;
END;
```

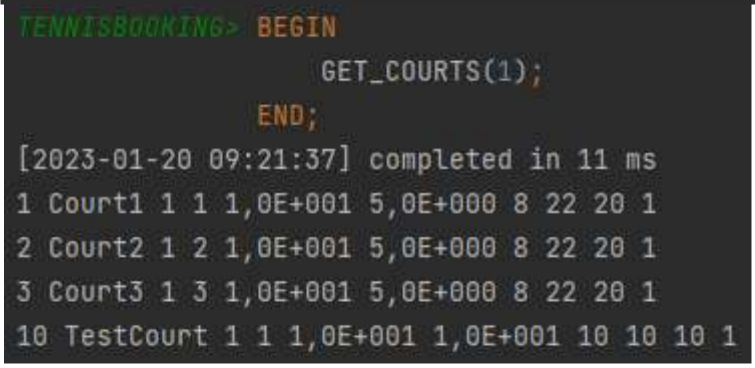


```
TENNISBOOKING> BEGIN
                   GET_CLUBS('TC');
               END;
[2023-01-20 09:21:01] completed in 13 ms
2 TC Eichgraben2 club2
3 TC Eichgraben3 club3
```

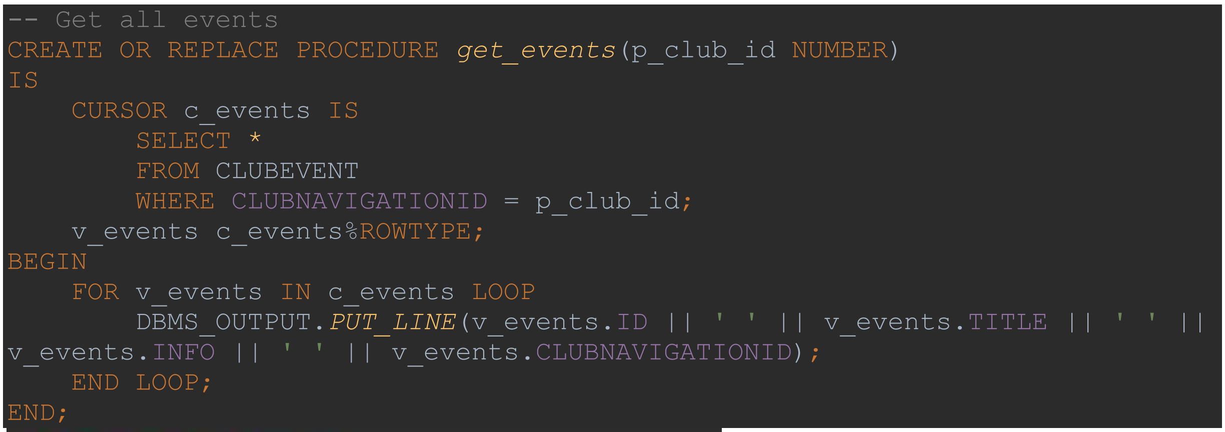
```
-- Get all courts
CREATE OR REPLACE PROCEDURE get_courts(p_club_id NUMBER)
IS
    CURSOR c_courts IS
        SELECT *
```



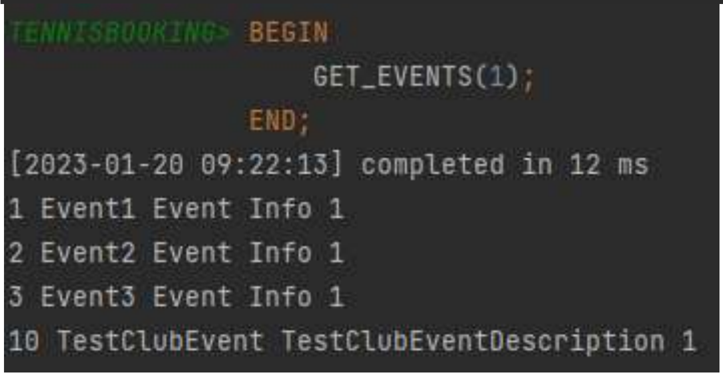
```
        FROM COURT
        WHERE CLUBNAVIGATIONID = p_club_id;
    v_courts c_courts%ROWTYPE;
BEGIN
    FOR v_courts IN c_courts LOOP
        DBMS_OUTPUT.PUT_LINE(v_courts.ID || ' ' || v_courts.NAME || ' ' ||
v_courts.BOOKABLE || ' ' || v_courts.TYPE || ' ' || v_courts.APRICE || ' '
|| v_courts.BPRICE || ' ' || v_courts.ATIMEFROM || ' ' ||
v_courts.ATIMETILL || ' ' || v_courts.AWEEKENDTIMETILL || ' ' ||
v_courts.CLUBNAVIGATIONID);
    END LOOP;
END;
```



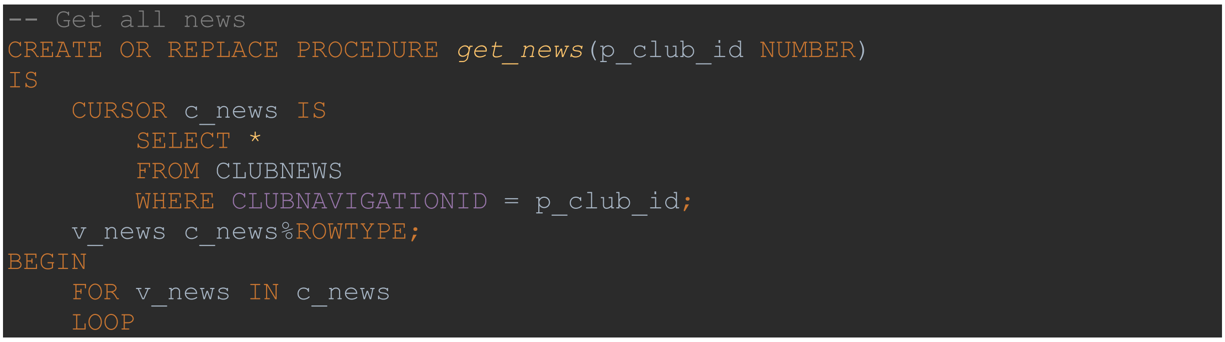
```
TENNISBOOKING> BEGIN
                   GET_COURTS(1);
               END;
[2023-01-20 09:21:37] completed in 11 ms
1 Court1 1 1 1,0E+001 5,0E+000 8 22 20 1
2 Court2 1 2 1,0E+001 5,0E+000 8 22 20 1
3 Court3 1 3 1,0E+001 5,0E+000 8 22 20 1
10 TestCourt 1 1 1,0E+001 1,0E+001 10 10 10 1
```



```
-- Get all events
CREATE OR REPLACE PROCEDURE get_events(p_club_id NUMBER)
IS
    CURSOR c_events IS
        SELECT *
        FROM CLUBEVENT
        WHERE CLUBNAVIGATIONID = p_club_id;
    v_events c_events%ROWTYPE;
BEGIN
    FOR v_events IN c_events LOOP
        DBMS_OUTPUT.PUT_LINE(v_events.ID || ' ' || v_events.TITLE || ' ' ||
v_events.INFO || ' ' || v_events.CLUBNAVIGATIONID);
    END LOOP;
END;
```

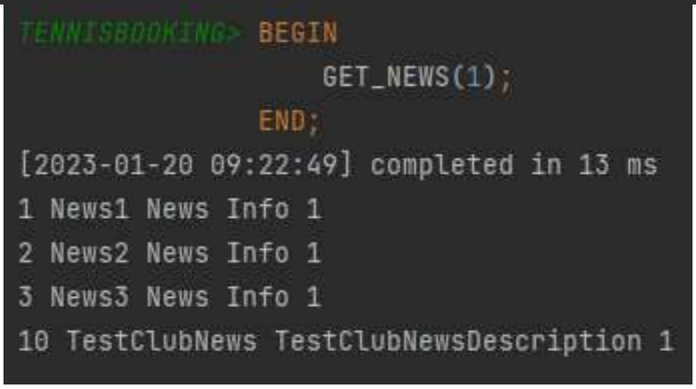


```
TENNISBOOKING> BEGIN
                   GET_EVENTS(1);
               END;
[2023-01-20 09:22:13] completed in 12 ms
1 Event1 Event Info 1
2 Event2 Event Info 1
3 Event3 Event Info 1
10 TestClubEvent TestClubEventDescription 1
```



```
-- Get all news
CREATE OR REPLACE PROCEDURE get_news(p_club_id NUMBER)
IS
    CURSOR c_news IS
        SELECT *
        FROM CLUBNEWS
        WHERE CLUBNAVIGATIONID = p_club_id;
    v_news c_news%ROWTYPE;
BEGIN
    FOR v_news IN c_news
    LOOP
```

```
        DBMS_OUTPUT.PUT_LINE(v_news.ID || ' ' || v_news.TITLE || ' ' || v_news.INFO || ' ' || v_news.CLUBNAVIGATIONID);
    END LOOP;
END;
```



```
TENNISBOOKING> BEGIN
                   GET_NEWS(1);
               END;
[2023-01-20 09:22:49] completed in 13 ms
1 News1 News Info 1
2 News2 News Info 1
3 News3 News Info 1
10 TestClubNews TestClubNewsDescription 1
```