



پروژه سوم درس پایگاه داده

استاد درس: دکتر شهریاری

محمد حسین شهبازی

۴۰۰۳۱۰۸۸

نتیجه اجرای پروسه ها (Procedures)

۱. پروسه‌های بنویسید که زوج نوع و تعداد را برای غذاهای رزرو شده امروز نشان دهد.

کد نوشته شده:

```
"1": {
  "name": "Create get_food_reservation_counts_today",
  "sql": ""
}

CREATE OR REPLACE PROCEDURE get_food_reservation_counts_today(
  INOUT result_set refcursor
)
LANGUAGE plpgsql
AS $$
BEGIN
  OPEN result_set FOR
  SELECT
    f.name AS food_name,
    COUNT(r.id) AS reservation_count
  FROM
    public.foods f
  JOIN
    public.reservations r ON f.id = r.foodid
  WHERE
    f.date::date = CURRENT_DATE
  GROUP BY
    f.name;
END;
$$;
```

توضیح: این پروسه می آید و با ترکیب دو جدول رزرو ها و غذا ها روی شناسه غذا ، تعداد غذا های که به تاریخ امروز می باشند را بر می گرداند.

خروجی تولید شده:

```
Choose an option: 4
1. CALL get_food_reservation_counts_today('result_set');
2. CALL get_students_with_reservation_today('result_set');
3. CALL get_recent_transactions('result_set');
4. CALL get_daily_revenue();
5. CALL calculate_remaining_inventory('result_set');
Choose a procedure to run: 1
('kabab', 2)
('salad_sezar', 1)
('takou', 1)
```

۲. پروسه ای بنویسید که لیست افرادی که امروز غذا رزرو کرده اند را نشان دهد.

کد نوشته شده:

```
"2": {
  "name": "Create get_students_with_reservation_today",
  "sql": ""
}

CREATE OR REPLACE PROCEDURE get_students_with_reservation_today(
  INOUT result_set refcursor
)
LANGUAGE plpgsql
AS $$
BEGIN
  OPEN result_set FOR
  SELECT
    *
  FROM
    public.students s
  JOIN
    public.reservations r ON s.studentid = r.studentid
  JOIN
    public.foods f ON r.foodid = f.id
  WHERE
    f.date::date = CURRENT_DATE;
END;
$$;
```

توضیح: این کد می‌آید و با ترکیب سه جدول دانشجوی و رزرو ها و غذا ها روی شناسه دانشجوی و شناسه غذا، مشخصات دانشجویانی که و رزرو ی که در تاریخ امروز انجام داده اند را بر می‌گرداند.

خروجی تولید شده:

```
Choose an option: 1
1. CALL get_food_reservation_counts_today('result_set');
2. CALL get_students_with_reservation_today('result_set');
3. CALL get_recent_transactions('result_set');
4. CALL get_daily_revenue();
5. CALL calculate_remaining_inventory('result_set');
Choose a procedure to run: 3
(7, 200, 'art', datetime.date(2010, 1, 1), 'ali', 'ahmadi', 360, 9, 200, 10, 10, 'kabab', datetime.datetime(2024, 7, 3, 0, 12, 17), 80, 6)
(8, 300, 'art', datetime.date(2011, 2, 1), 'erfan', 'saffari', 370, 10, 300, 11, 11, 'takou', datetime.datetime(2024, 7, 3, 0, 13, 17), 70, 6)
(8, 300, 'art', datetime.date(2011, 2, 1), 'erfan', 'saffari', 370, 11, 300, 12, 12, 'salad-sezan', datetime.datetime(2024, 7, 3, 0, 15, 17), 60, 5)
(5, 100, 'cs', datetime.date(2010, 1, 1), 'mohammad', 'rahmani', 360, 12, 100, 10, 10, 'kabab', datetime.datetime(2024, 7, 3, 0, 12, 17), 80, 6)
```

۳. پروسه‌های بنویسید که لیستی از ۱۰ تراکنش اخیر سیستم را نشان دهد.

کد نوشته شده:

```
"3": {
    "name": "Create get_recent_transactions",
    "sql": ""
}

CREATE OR REPLACE PROCEDURE get_recent_transactions(
    INOUT result_set refcursor
)
LANGUAGE plpgsql
AS $$
BEGIN
    OPEN result_set FOR
    SELECT
        srcreservationid,
        dstreservationid,
        date
    FROM
        public.transactions
    ORDER BY
        date DESC
    LIMIT 10;
END;
$$;
""
```

توضیح: این پروسه روی جدول رزرو ها اعمال شده و خروجی روی محدودیت ۱۰ ردیف تولید می شود و بر گردانده می شود.

خروجی تولید شده:

```
Choose a procedure to run: 3
(None, 10, datetime.datetime(2024, 7, 3, 0, 15, 31, 755710))
(None, 12, datetime.datetime(2024, 7, 3, 0, 15, 23, 735948))
(None, 11, datetime.datetime(2024, 7, 3, 0, 15, 8, 509251))
(None, 10, datetime.datetime(2024, 7, 3, 0, 14, 54, 95457))
(None, 6, datetime.datetime(2024, 7, 3, 0, 5, 2, 202721))
(10, 11, datetime.datetime(2024, 4, 22, 10, 54, 30, 809391))
(4, 6, datetime.datetime(2024, 4, 22, 10, 54, 30, 809391))
(7, 8, datetime.datetime(2024, 4, 22, 10, 54, 30, 809391))
(None, 4, datetime.datetime(2024, 4, 22, 10, 52, 32, 698582))
```

۴. پروسهای بنویسید که زوج نوع و تعداد را برای غذاهای باقی مانده امروز نمایش دهد. غذاهایی که تعداد رزروهایشان کمتر از inventory آنها باشد.

کد نوشته شده:

```
name : create_calculate_remaining_inventory ;
"sql": ""
CREATE OR REPLACE PROCEDURE calculate_remaining_inventory(
  INOUT result_set refcursor
)
LANGUAGE plpgsql
AS $$
BEGIN
  OPEN result_set FOR
  SELECT
    f.name AS food_name,
    (f.inventory - COALESCE(s.reservation_count, 0)) AS remaining_count
  FROM
    public.foods f
  LEFT JOIN (
    SELECT
      r.foodid,
      COUNT(r.id) AS reservation_count
    FROM
      public.reservations r
    GROUP BY
      r.foodid
  ) s ON f.id = s.foodid;
END;
$$;
""
```

Font size: 11pt

توضیح : این پروسه می‌آید و ردیف هایی را بر می‌گرداند که تعداد رزرو هایشان از تعداد سفارش شده آن ها کمتر باشد. در اینجا نوع غذا نیز در نظر گرفته شده است.

خروجی تولید شده:

```
Choose a procedure to run: 5
('salad1', 10)
('soup1', 5)
('khorest', 6)
('takou', 5)
('salad_sezar', 4)
('kabab', 4)
('salad_3', 600)
('salad_4', 600)
```

۵. پروسهای بنویسید که گردش مالی روزانه سلف دانشگاه را نشان دهد. مجموع قیمت غذاهای رزرو شده برای یک روز.

کد نوشته شده:

```
"name": "Create get_daily_revenue",
"sql": """
        CREATE OR REPLACE FUNCTION get_daily_revenue()
        RETURNS DECIMAL AS $$
        DECLARE
            daily_revenue DECIMAL;
        BEGIN
            SELECT SUM(f.price)
            INTO daily_revenue
            FROM public.reservations r
            JOIN public.foods f ON r.foodid = f.id
            WHERE f.date::date = CURRENT_DATE;

            RETURN daily_revenue;
        END;
        $$ LANGUAGE plpgsql;"""
```


توضیح : این کد می آید و با ترکیب دو جدول غذا ها و رزرو ها روی شناسه غذا، قیمت غذا های به تاریخ امروز سفارش داده شده را بر می گرداند.

خروجی تولید شده:

```
Choose an option: 7
Daily Revenue: 290
```

خروجی اجرای تریگر ها (Triggers):

۱. اضافه شدن دانشجو با کد ملی یا دانشجویی یکسان. نتیجه:

```
1 INSERT INTO foods (name, date, price, inventory)
2 VALUES ('salad_3', '2024-10-03 01:00:00', 200, 10);
3
```

Data Output

Messages

Notifications

ERROR: initial inventory for food salad_3 must be at least 500
CONTEXT: PL/pgSQL function check_initial_inventory() line 4 at RAISE
SQL state: P0001

۲. کمتر از ۵۰۰ بودن موجودی اولیه هر غذا.

نتیجه:

```
1 INSERT INTO foods (name, date, price, inventory)
2 VALUES ('salad_3', '2024-10-03 01:00:00', 200, 10);
3
```

Data Output Messages Notifications

ERROR: initial inventory for food salad_3 must be at least 500
CONTEXT: PL/pgSQL function check_initial_inventory() line 4 at RAISE

SQL state: P0001

۳. سرو شدن بالای ۲ نوع غذا در یک زمان.

نتیجه :

Query Query History

```
1 insert into foods (name , date , price , inventory) values ('salad_5' , '2024-07-03 00:13:17' , 800 , 600)
```

Data Output Messages Notifications

ERROR: More than two foods cannot have the same date
CONTEXT: PL/pgSQL function check_multiple_foods_same_date() line 4 at RAISE
SQL state: P0001

۴. سفارش بیش از یک غذا توسط یک دانشجو در یک زمان.

نتیجه:

Query	Query History
-------	---------------

```
1 insert into reservations (studentid , foodid) values (100,6)
```

Data Output	Messages	Notifications
-------------	----------	---------------

ERROR: Student 100 cannot order more than one food item simultaneously.
CONTEXT: PL/pgSQL function check_multiple_students_order() line 9 at RAISE

SQL state: P0001

پیاده سازی ویو (View) :

۱. یک view با نام food_student تعریف کنید که نام همه غذاهای رزرو شده توسط دانشجو را در کنار نام او نشان دهد.

خروجی query :

Query

Query History










1

select * from food_student

Data Output

Messages

Notifications



	student_name text	food_name character varying (50)
1	mohammad rahmani	khorest
2	ali ahmadi	khorest
3	ali ahmadi	kabab
4	erfan saffari	takou
5	erfan saffari	salad_sezar
6	mohammad rahmani	kabab

۲. یک view با نام transaction_student بنویسید که در کنار نام دانشجو و شماره دانشجویی اش، اطلاعات رزروی که تغییر داده است شامل نام غذای رزرو اولیه و غذای رزرو جدید ذکر شود.

Query

Query History

1 select * from transaction_student

Data Output

Messages

Notifications

≡+

▼

▼

	studentid integer	student_name text	original_food character varying (50)	new_food character varying (50)
1	100	mohammad rahmani	khorest	khorest
2	300	erfan saffari	takou	salad_sezar

جلوگیری از ورود داده های نامعتبر:

خروجی:

Query

Query History

1

```
insert into transactions (srcreservationid , dstreservationid ,date ) values(4 , 6 , '2024-05-22 10:52:32.698582' )
```

Data Output

Messages

Notifications

ERROR: Key (srcreservationid, dstreservationid)=(4, 6) already exists.duplicate key value violates unique constraint "unique_transaction_ids"

ERROR: duplicate key value violates unique constraint "unique_transaction_ids"

SQL state: 23505

Detail: Key (srcreservationid, dstreservationid)=(4, 6) already exists.

نحوه بهبود پردازش کوئریها و مقایسه عملکرد حاصل:

کد نوشته شده:

```
def create_index_and_compare(conn):  
    initial_query = "SELECT * FROM public.students WHERE studentid = 1001;"  
    index_query = "CREATE INDEX idx_studentid ON public.students (studentid);"  
  
    initial_time = time_query_execution(conn, initial_query)  
    print(f"Initial execution time: {initial_time:.6f} seconds")  
  
    execute_query(conn, index_query)  
  
    final_time = time_query_execution(conn, initial_query)  
    print(f"Execution time after indexing: {final_time:.6f} seconds")  
  
    time_difference = initial_time - final_time  
    print(f"Difference in execution time: {time_difference:.6f} seconds")
```

توضیح: در اینجا یک index query نوشته شده و تفاوت زمانی حالت عادی و حالت جستجو در index را نشان می‌دهد.

خروجی تولید شده:

```
Choose an option: 5  
Initial execution time: 0.012000 seconds  
query executed successfully.  
Execution time after indexing: 0.001001 seconds  
Difference in execution time: 0.010999 seconds
```