Requesting a report:

```
    Insert into report_details (user_id, time_requested, report_description) values
    (x,now(),x);
```

Most expensive meal:

```
select ans1.meal_id,ans2.price
from
    (select MAX(ans1.price) price
    from (
        select sum(mpp.part_price) price
        from meal m, (
            select mm.meal_id,(ing.price*m_ing.quantity) part_price
            from meal_ingredients m_ing, ingredient ing, meal mm
            where (m_ing.meal_id=mm.meal_id and
ing.ingredient_id=m_ing.ingredient_id)) mpp
        where m.meal_id=mpp.meal_id
        group by m.meal_id) ans1) ans2
    inner join
        (
        select m.meal_id,sum(mpp.part_price) price
        from meal m, (
            select mm.meal_id,(ing.price*m_ing.quantity) part_price
            from meal_ingredients m_ing, ingredient ing, meal mm
            where (m_ing.meal_id=mm.meal_id and
ing.ingredient_id=m_ing.ingredient_id)) mpp
        where m.meal_id=mpp.meal_id
        group by m.meal_id) ans1
    on ans1.price=ans2.price;
```

Most ordered meal

```
select meal_id,ord_times
from (
    select m.meal_id, count(ord.*) ord_times
    from meal m, orders ord
    where m.meal_id=ord.meal_id
    group by m.meal_id) with_id
inner join(
    select max(ords.ord_times) max_ord
    from(
        select count(ord.*) ord_times
        from meal m, orders ord
        where m.meal_id=ord.meal_id
        group by m.meal_id) ords) without_id
on without_id.max_ord=with_id.ord_times;
```

Top 3 used ingredients

```
    Select ing.name, count(*) as num
    From ingredient ing, meal_ingredients m_ing
    Where ing.ingredient_id=m_ing.ingredient_id
    Group by ing.ingredient_id
    order by num desc
    Limit 3;
```

Add New Meal:

```
    INSERT INTO
    meal(name,cuisine_id,description,time_added,time_modified,avd_rate,num_rates)
    VALUES (x,x,x,now(),now(),x,x);
```

Select a meal given Meal Id (@mealid):

```
    SELECT * FROM meal WHERE meal_id = @mealid
```

Add New Ingredient:

```
    INSERT INTO ingredient(name, category, quantity, unit, price, order_threshold,
    is_requested, time_added, time_modified)
    VALUED (x,x,x,x,x,x,x,now(),now());
```

Select an Ingredient given Ingredient id (@ingredient_id):

```
    SEIECT * FROM ingredient WHERE ingredient_id = @ingredient_id
```

Add new Cuisine Type:

```
    INSERT INTO cuisine_type(name, country_of_origin, description, time_added,
    time_modified)
    VALUES (x,x,x,now(), now())
```

Select a Cuisine type given Cuisine id (@cuisine_id)

```
    SELECT * FROM cuisine_type WHERE cuisine_id = @cuisine_id
```

Add new request:

```
INSERT INTO request(is_fulfilled, time_added, time_modified, amount)
VALUES (x,now(),now(),x);
```

Select all requests and order by oldest first:

```
SELECT * FROM request ORDER BY time_added DESC;
```

Generating reports

```
CREATE OR REPLACE FUNCTION GenerateReport(id INT,desc VARCHAR(200))
RETURNS VOID AS $$
BEGIN
        INSERT INTO REPORT_DETAILS
        (user_id,time_requested,report_description)
        VALUES(id, now(), desc)
END
```

Average Rating:

```
SELECT AVG(score) FROM RATING
WHERE meal_id = @id
```

Count of Ratings:

```
SELECT COUNT(score) FROM RATING
WHERE meal_id = @id
```

Stored Procedure user access

```
CREATE OR REPLACE FUNCTION GenerateReport(id INT,desc VARCHAR(200))
RETURNS VOID AS $$
BEGIN
        INSERT INTO REPORT_DETAILS
        (user_id,time_requested,time_modified,query_description)
        VALUES(id, now(),now(), desc)
END
```