Project_0506_12_Design

DESIGN

Brand Name: The Terrapin Cuisine Crew

Business Processes/Transactions in sentences:
Each reviewer is described by a unique Id and has its Name, Number.
Each restaurant is described by a unique ID, name, address (consists of Street, City, State and Zip) restaurant type, the established year operation and the phone number of it.
Reviews are not independent of reviewers, each review is identified by a unique review Id, Type, Date, Star and Source.
Each cuisine is described by a unique Id and Type.
Each order is described by a unique Id and ItemName.
Each reviewer can review one or more restaurants.
Each restaurant can be reviewed by one or more reviewers.
Each restaurant could offer one or many cuisines.
Each cuisine can only be offered by one restaurant.
The amenities in each restaurant are described by a unique ID

Four Entity Types: Reviewer, Restaurant, Cuisine, Features
Relationship Types: Many to many, many to one, many to one
ER Schema: In proposal
ER Diagram: In proposal

PROPOSAL

Mission statement :

-To analyze online reviews of restaurants in College Park, MD in order to gain insights on restaurant ratings and customer experiences.

-To collect, sort, and analyze the comment focuses of online restaurant reviews in order to assist database users in ordering from College Park, MD owned restaurants based on different dimensions

Mission objectives :

-To find the positive and negative feedback rates of each restaurant (4 points or more is considered positive)

-To find the restaurants with the the most reviews in 2021 among these ten restaurants

Project_0506_12_Design

-To find the restaurant that offer the most features in College Park, MD

-To find the average rating of each restaurant

- To find which aspects of each restaurant are rated positively and negatively

ER Schema:

Entities, Attributes and Primary Keys

Reviewer (**revrId**, revrName, revrEmail, revrDate, revrText, revrStar, revrSource, revrExperience)

Restaurant(**restId**, restName, restAdress, restStreet, restCity, restState, restZip, restEstabYear, restOpenTime, restCloseTime, restPhone)

Cuisine (**cuisId**, cuisType)

Features(**FeaId**, FeaReservation, FeaDeliver, FeaTakeOut)

Relationships, Attributes, Degrees, Participating Entities and Constraints

Review: Binary relationship

1 reviewer to 1 or more restaurants

1 restaurant to 1 or more reviewers

Offer: Binary relationship

1 restaurant to 1 or more cuisine

1 cuisine to 1 restaurant

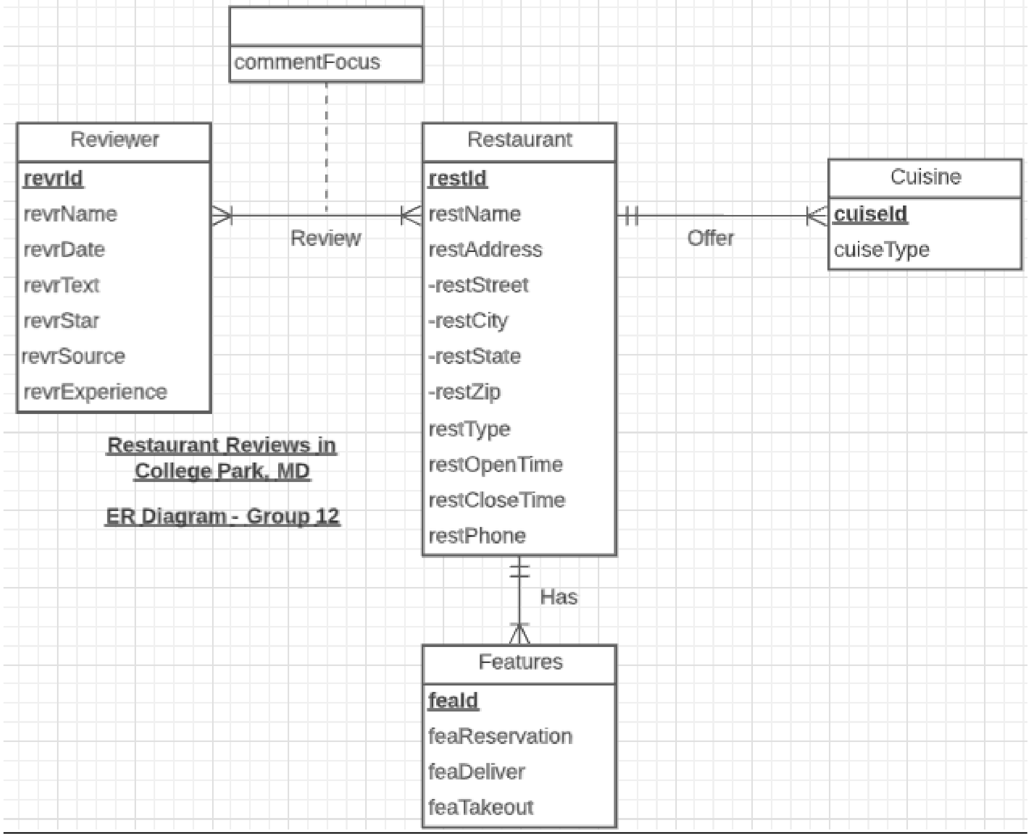
Has: Binary relationship

1 restaurant to 1 or more features

1 feature to 1 restaurant

ER Diagram - LucidChart (Updated)

Project_0506_12_Design



Relation schema with PK& FK

Relations:

Reviewer (**<u>revrId</u>**, revrName, revrDate, revrText, revrStar, revrSource, revrExperience, *restId*)

Restaurant (**<u>restId</u>**, restName, restStreet, restCity, restState, restZip, restOpenTime, restCloseTime, restPhone)

Cuisine (**<u>cuisId</u>**, cuisType, *restId*)

Features(**<u>feaId</u>**, feaReservation, feaDeliver, feaTakeout, *restId*)
Review (***<u>restId, revrId</u>***, commentFocus)

Functional dependencies:

Business rules :

[R1] When a restaurant is deleted from or changed in the database, all cuisine's for the restaurant should be deleted from the database or change accordingly as well

[R2] When a restaurant is deleted from or changed in the database, all features for the restaurant should be deleted from the database or changed accordingly as well

[R3] When there is a review done by a reviewer on a restaurant, the restaurant and the reviewer cannot be changed or deleted in the database.

Referential integrity actions:

Relation	FK	Base Relation	PK	Business Rule	Constraint on DELETE	Business Rule	Constraint on UPDATE
Features	restId	Restaurant	restId	R2	CASCADE	R2	CASCADE
Cuisine	restId	Restaurant	restId	R1	CASCADE	R1	CASCADE
Review	revrId	Reviewer	revrId	R3	NO ACTION	R3	NO ACTION
Review	restId	Restaurant	restId	R3	NO ACTION	R3	NO ACTION

Sample data description for every relation:

Reviewer (V001, 'Whitney L.', 9/6/2020, 'Standard Chick-fil-A quality food but I've heard mixed reviews about the employees at this establishment. Some have been known to be very friendly while others are more stand-offish. With COVID going on, just know that you do require an appointment to access the STAMP building in order to grab food.', '3 Stars', 'N', 'Yelp')

Cuisine (C001, 'American', R001)

Features(F001, 'Y', 'Y', 'N', R001)

Restaurant(R001, ‘Chick fil-a’, ‘3972 Campus Dr‘, ‘College Park’,‘MD’,’20740’, ‘9:00AM’, ‘12:00PM’, ‘(301)314-6568’, ’Nuggets’)

DDL

SQL Create/Drop Table, Create/Drop View, Insert Into, Alter Table Statements

```
USE BUDT703_DB_Student_184

DROP TABLE IF EXISTS [G12.Review]

DROP TABLE IF EXISTS [G12.Features]

DROP TABLE IF EXISTS [G12.Cuisine]

DROP TABLE IF EXISTS [G12.Restaurant]

DROP TABLE IF EXISTS [G12.Reviewer]

CREATE TABLE [G12.Reviewer] (

        revrId CHAR(10) NOT NULL,

        revrName VARCHAR(20),

        revrEmail VARCHAR(20),

        revrDate DATE,

        revrText VARCHAR(800),

        revrStar INTEGER,

        revrSource VARCHAR(20),

        restId CHAR(10) NOT NULL,

        reverExperience CHAR(10),

        CONSTRAINT pk_Reviewer_revrId PRIMARY KEY (revrId))

        CONSTRAINT fk_Reviewer_restId FOREIGN KEY (restId)

                REFERENCES [G12.Restaurant](restId)

                ON DELETE CASCADE ON UPDATE CASCADE)



INSERT INTO [G12.Reviewer] (revrId, revrName, revrEmail, revrDate, revrText, revrStar, revrSource)

        VALUES ('V000000001', 'Whitney L.', 'Whitney@umd.edu', '9/6/2020', 'Standard Chick-fil-A
quality food but Ive heard mixed reviews about the employees at this establishment. Some have been
known to be very friendly while others are more stand-offish. With COVID going on, just know that you
do require an appointment to access the STAMP building in order to grab food.', '3', 'Yelp');

CREATE TABLE [G12.Restaurant] (

        restId CHAR(10) NOT NULL,
        restName VARCHAR(20),

        restStreet VARCHAR(20),

        restCity VARCHAR(20),

        restState CHAR(2),

        restZip CHAR(5),

        restOpenTime DATETIME(8),

        restCloseTime DATETIME(8),

        restEstabYear CHAR(4),

        restPhone CHAR(10),

        CONSTRAINT pk_Restaurant_restId PRIMARY KEY (restId))

INSERT INTO [G12.Restaurant] (restId, restName, restStreet, restCity, restState, restZip,
restEstabYear, restPhone)

        VALUES ('R100000000', 'Chick fil-a', '3972 Campus Dr', 'College Park', 'MD', '20740', '1946',
'3103146568');

CREATE TABLE [G12.Cuisine] (

        cuisId CHAR(4),

        cuisType VARCHAR(20),

        restId CHAR(10),

        CONSTRAINT pk_Cuisine_cuisId PRIMARY KEY (cuisId),

        CONSTRAINT fk_Cuisine_restId FOREIGN KEY (restId)

                REFERENCES [G12.Restaurant](restId)

                ON DELETE CASCADE ON UPDATE CASCADE)

INSERT INTO [G12.Cuisine] (cuisId, cuisType, restId)

        VALUES ('C001', 'American', 'R100000000');

CREATE TABLE [G12.Features] (

        feaId CHAR(4),

        feaReservation CHAR(1),

        feaDeliver CHAR(1),

        feaTakeOut CHAR(1),

        restId CHAR(10),

        CONSTRAINT pk_Features_feaId PRIMARY KEY (feaId),

        CONSTRAINT fk_Features_restId FOREIGN KEY (restId)

                REFERENCES [G12.Restaurant] (restId)
                ON DELETE CASCADE ON UPDATE CASCADE)

INSERT INTO [G12.Features] (feaId, feaReservation, feaDeliver, feaTakeOut, restId)

        VALUES ('F000', 'N', 'Y', 'N', 'R100000000' );


CREATE TABLE [G12.Review] (

        restId CHAR(10),

        revrId CHAR(10),

        CommentFocus VARCHAR(20),

        CONSTRAINT pk_Review_restId_revrId PRIMARY KEY (restId, revrId),

        CONSTRAINT fk_Review_restId FOREIGN KEY (restId)

                REFERENCES [G12.Restaurant] (restId)

                ON DELETE NO ACTION ON UPDATE NO ACTION,

        CONSTRAINT fk_Review_revrId FOREIGN KEY (revrId)

                REFERENCES [G12.Reviewer] (revrId)

                ON DELETE NO ACTION ON UPDATE NO ACTION)

INSERT INTO [G12.Review] (restId, revrId, ordItemName)

        VALUES ('R100000000', 'V000000001',  'Chicken Tenders');
```

DML

Identify at least four business transactions, in WH-questions, to implement mission objectives.

-To find the positive/negative feedback rate of each restaurant (4 points or more is considered positive)

-To find the restaurants with the the most reviews in 2021 among these ten restaurants

-To find the restaurant with the most features in College Park, MD

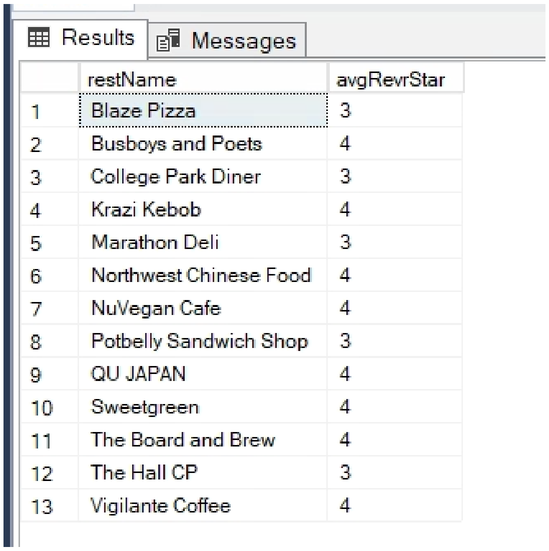
-To find the average rate of each restaurant

Implement SQL SELECT and CREATE VIEW statements to answer business transactions.

```
use BUDT703_Project_0506_12;
DROP VIEW IF EXISTS V1
DROP VIEW IF EXISTS V2
```

Project_0506_12_Design

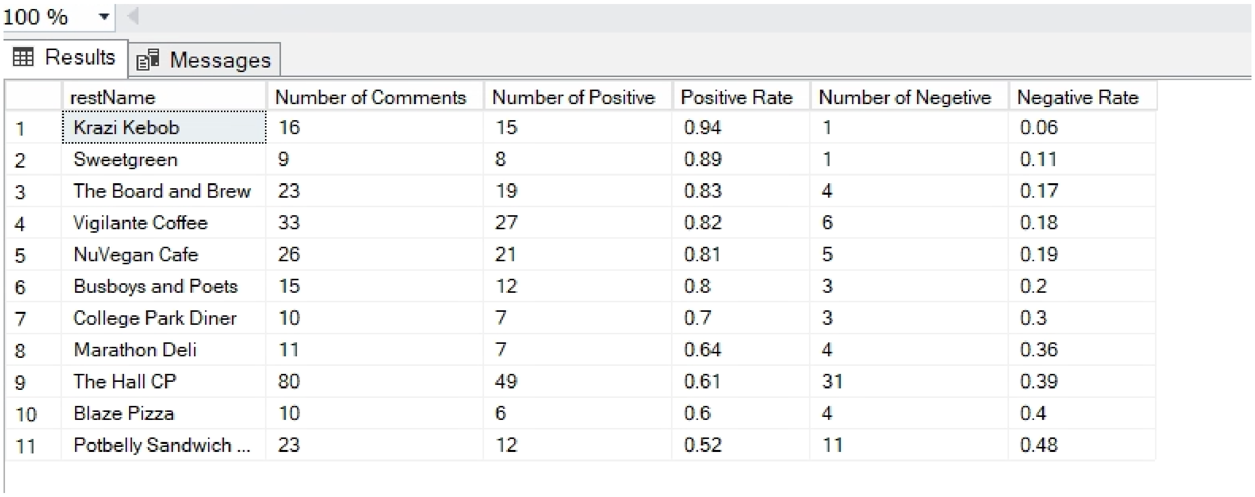
```
DROP VIEW IF EXISTS V3
DROP VIEW IF EXISTS V4
DROP VIEW IF EXISTS focus
DROP VIEW IF EXISTS V5

GO
CREATE VIEW V1
AS
--Q1: What the average star rate of each restaurant
SELECT r.restName,AVG(r.revrStar) 'avgRevrStar'
FROM [G12.Reviewer] r
GROUP BY r.restName
```



	restName	avgRevrStar
1	Blaze Pizza	3
2	Busboys and Poets	4
3	College Park Diner	3
4	Krazi Kebob	4
5	Marathon Deli	3
6	Northwest Chinese Food	4
7	NuVegan Cafe	4
8	Potbelly Sandwich Shop	3
9	QU JAPAN	4
10	Sweetgreen	4
11	The Board and Brew	4
12	The Hall CP	3
13	Vigilante Coffee	4

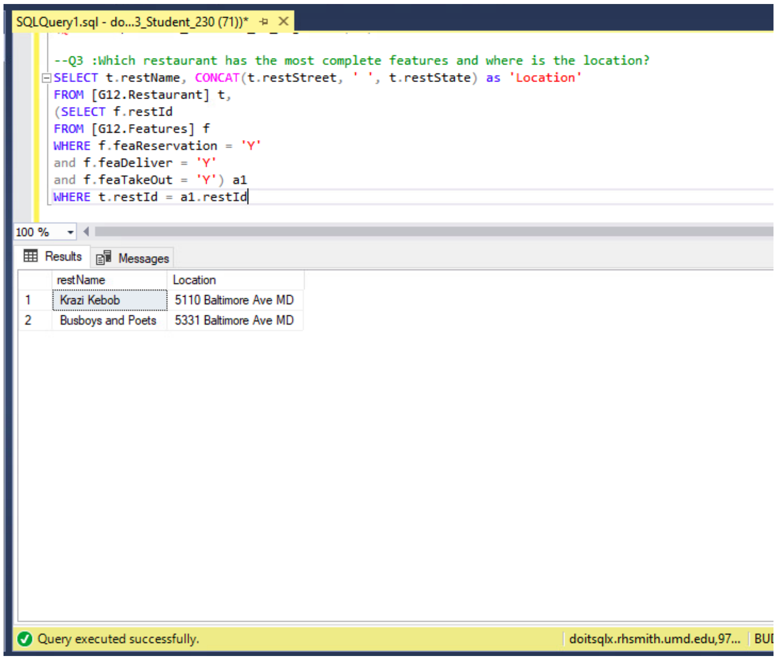
```
--Q2: How many negative/positive reviews each restaurant received? And what about their rate? order by the rate of
positive?
GO
CREATE VIEW V2
AS
SELECT r.restName, COUNT(r.revrExperience_Positive_or_Negative ) 'Number of Comments',
c1.Pos 'Number of Positive', ROUND(CAST(c1.Pos AS FLOAT)/COUNT
(r.revrExperience_Positive_or_Negative ),2)'Positive Rate', c1.Neg 'Number of Negative',
ROUND(CAST(c1.Neg AS FLOAT) /COUNT (r.revrExperience_Positive_or_Negative ),2) 'Negative Rate'
FROM [G12.Reviewer] r, (
            SELECT n.restName, n.Neg, p.Pos
            FROM (
                        SELECT restName, COUNT(revrExperience_Positive_or_Negative ) 'Neg'
                        FROM [G12.Reviewer]
                        WHERE revrExperience_Positive_or_Negative = 'N'
                        GROUP BY restName) n, (
                                    SELECT restName, COUNT(revrExperience_Positive_or_Negative ) 'Pos'
                                    FROM [G12.Reviewer]
                                    WHERE revrExperience_Positive_or_Negative = 'P'
                                    GROUP BY restName) p
            WHERE n.restName=p.restName) c1
WHERE r.restName = c1.restName
```

```
GROUP BY r.restName, c1.Pos,c1.Neg
```



	restName	Number of Comments	Number of Positive	Positive Rate	Number of Negetive	Negative Rate
1	Krazi Kebob	16	15	0.94	1	0.06
2	Sweetgreen	9	8	0.89	1	0.11
3	The Board and Brew	23	19	0.83	4	0.17
4	Vigilante Coffee	33	27	0.82	6	0.18
5	NuVegan Cafe	26	21	0.81	5	0.19
6	Busboys and Poets	15	12	0.8	3	0.2
7	College Park Diner	10	7	0.7	3	0.3
8	Marathon Deli	11	7	0.64	4	0.36
9	The Hall CP	80	49	0.61	31	0.39
10	Blaze Pizza	10	6	0.6	4	0.4
11	Potbelly Sandwich ...	23	12	0.52	11	0.48

```
--Q3 :Which restaurant has the most complete features and where is the location?
GO
CREATE VIEW V3
AS
SELECT t.restName, CONCAT(t.restStreet, ' ', t.restState) as 'Location'
FROM [G12.Restaurant] t,
(SELECT f.restId
FROM [G12.Features] f
WHERE f.feaReservation = 'Y'
and f.feaDeliver = 'Y'
and f.feaTakeOut = 'Y') a1
WHERE t.restId = a1.restId
```



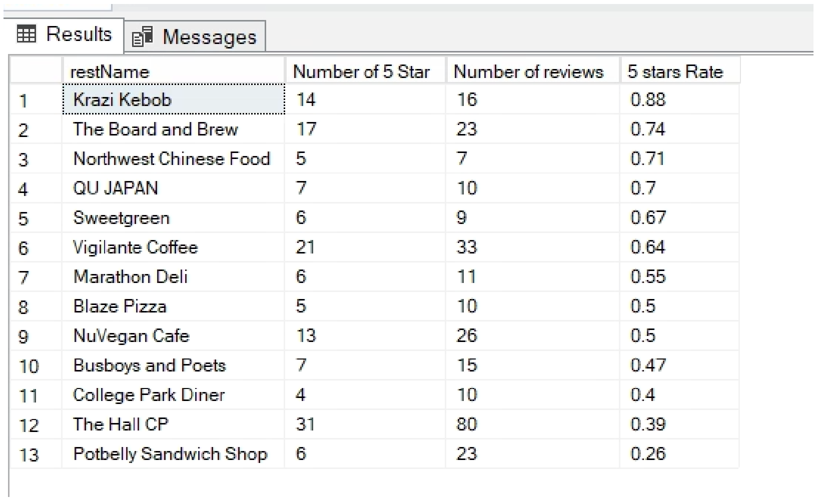
```
--Q3 :Which restaurant has the most complete features and where is the location?
SELECT t.restName, CONCAT(t.restStreet, ' ', t.restState) as 'Location'
FROM [G12.Restaurant] t,
(SELECT f.restId
FROM [G12.Features] f
WHERE f.feaReservation = 'Y'
and f.feaDeliver = 'Y'
and f.feaTakeOut = 'Y') a1
WHERE t.restId = a1.restId
```

	restName	Location
1	Krazi Kebob	5110 Baltimore Ave MD
2	Busboys and Poets	5331 Baltimore Ave MD

Query executed successfully.

doitsqlx.rhsmith.umd.edu,97...

```
--Q4: What is the 5 Stars Rate of each restaurant ?
GO
CREATE VIEW V4
AS
SELECT c1.restName, c1.s AS 'Number of 5 Star', COUNT(e.revrId) AS 'Number of reviews', ROUND(CAST(c1.s as
FLOAT)/COUNT
(e.revrId),2) '5 stars Rate'
FROM [G12.Reviewer] e,(
            SELECT r.restName, COUNT(r.revrStar) s
            FROM [G12.Reviewer] r
            WHERE r.revrStar =5
            GROUP BY r.restName) c1
WHERE e.restName = c1.restName
GROUP BY c1.restName, c1.s
```



	restName	Number of 5 Star	Number of reviews	5 stars Rate
1	Krazi Kebob	14	16	0.88
2	The Board and Brew	17	23	0.74
3	Northwest Chinese Food	5	7	0.71
4	QU JAPAN	7	10	0.7
5	Sweetgreen	6	9	0.67
6	Vigilante Coffee	21	33	0.64
7	Marathon Deli	6	11	0.55
8	Blaze Pizza	5	10	0.5
9	NuVegan Cafe	13	26	0.5
10	Busboys and Poets	7	15	0.47
11	College Park Diner	4	10	0.4
12	The Hall CP	31	80	0.39
13	Potbelly Sandwich Shop	6	23	0.26

Project_0506_12_Design

–What is the distribution of reviews which are higher than 4 stars for each restaurant? The distribution is categorized within ‘environment/ food/ service’

```
CREATE VIEW focus AS
SELECT y1.restId,v.commentFocus,Y1.revrId
FROM [G12.Review] v ,(
        SELECT e.restId, e.revrId
        FROM [G12.Reviewer] e
        WHERE e.revrStar >3
GROUP BY e.restId, e.revrId) y1
WHERE y1.revrId =v.revrId

GO
CREATE VIEW V5
AS
SELECT g.restName, g.restId,ISNULL(sef.environment, 0) 'environment', ISNULL(sef.food,0) 'food',
ISNULL(sef.service,0) 'service'
FROM [G12.Restaurant] g, (
        SELECT ISNULL(f1.restId, se.restId) 'restId', f1.food, se.environment, se.service
        FROM (
                SELECT i.restId , COUNT(f.commentFocus) 'food'
                FROM [G12.Restaurant] i, focus f
                WHERE I.restId =f.restId
                GROUP BY i.restId, f.commentFocus
                HAVING f.commentFocus ='FOOD') f1
FULL JOIN (
                SELECT s.restId, s.service, e.environment
                FROM (
                        SELECT i.restId , COUNT(f.commentFocus) 'service'
                        FROM [G12.Restaurant] i, focus f
                        WHERE I.restId =f.restId
                        GROUP BY i.restId, f.commentFocus
                        HAVING f.commentFocus ='SERVICE') s
        FULL OUTER JOIN (
                SELECT i.restId , COUNT(f.commentFocus) 'environment'
                FROM [G12.Restaurant] i, focus f
                WHERE I.restId =f.restId
                GROUP BY i.restId, f.commentFocus
                HAVING f.commentFocus ='ENVIRONMENT') e
        ON s.restId=e.restId) se
ON se.restId = f1.restId) sef
WHERE g.restId=sef.restId
```