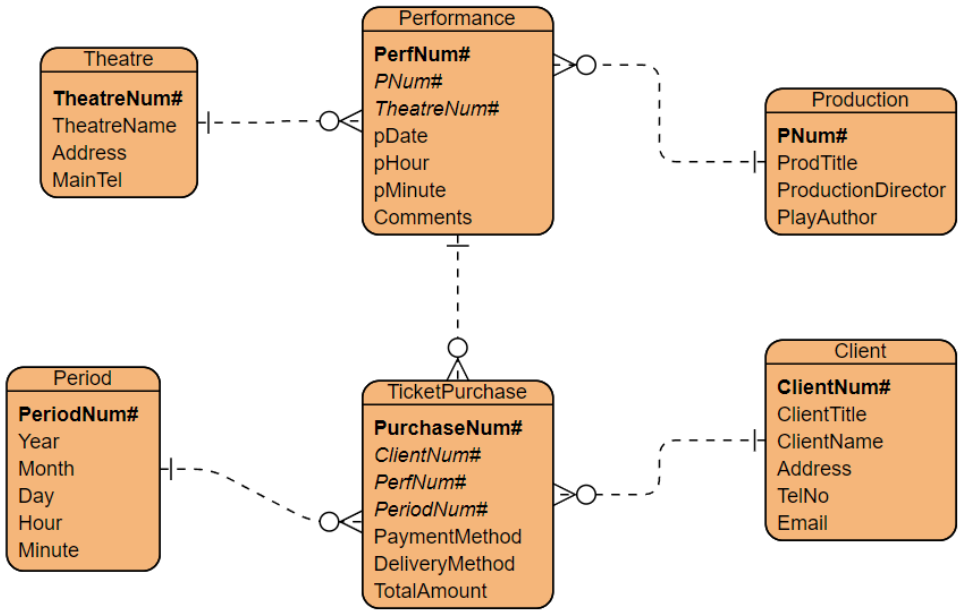
William Husband

21359557

6G6Z0026, 1CWK30

Data Governance and Management

Star Schema and Justification



Bold# = Primary Key, *Italic#* = Foreign Key

Manchester Theatre outline that they would like to be able to extract and analyse data in certain ways, like extracting data according to different time frames, extracting specific customer data, or extracting sales data. Adding a 'Period' dimension to the 'TicketPurchase' fact table, Manchester Theatre can easily extract data from different time frames. Plus, the granularity of the period table allows for precise analysis of this data. Other than this, the data mart design stays relatively loyal to the original design. The 'TotalAmount' measure allows for analysis of sales data, while the 'Client' dimension allows easy customer data to be easily extracted. The similarity to the previous layout could possibly ease current employees' migration to the new system. In future, this data mart could potentially be expanded into a data warehouse. Perhaps if Manchester Theatre decided to merge with another theatre company. The granularity and simplicity of this data mart would allow for straight forward data integration.

Screenshots of Create and Insert Statements

```sql
/*Script written by William Husband 21359557*/

/*dropping tables deletes any stored tables with the same name (for example from a previous run of this script) for a clean slate to start again so we don't get any issues with multiple tables with the same name */
DROP TABLE TicketPurchase;
DROP TABLE Performance;
DROP TABLE Client;
DROP TABLE Production;
DROP TABLE Period;
DROP TABLE Theatre;

/*creating Client table */
/*note: primary key is generated automatically, doesnt need to be inserted */
/*this table contains a primary key for ClientNum (client number)*/

/*DATA TYPES:*/
/*NUMBER(10) is used for keys as it allows up to 999,999,999 of distinct integers*/
/*VARCHAR2 is used for text data as it stores necessary amount of characters rather than just 1 with CHAR*/
/*DATE is used for bookings as it allows easy input of day, month, year data*/
/*DECIMAL is used for TotalAmount as it means we can perform arithmetic on it*/


CREATE TABLE Client (
    ClientNum      NUMBER(10) GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
    ClientTitle         VARCHAR2(50),
    ClientName     VARCHAR2(255) NOT NULL,
    Address        VARCHAR2(255),
    TelNo          VARCHAR2(15),
    Email          VARCHAR2(255)
);

/*creating Theatre table*/
/*this table contains a primary key for TheatrefNum (theatre number)*/
CREATE TABLE Theatre (
    TheatreNum     NUMBER(10) GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
    TheatreName          VARCHAR2(255) NOT NULL,
    Address        VARCHAR2(255) NOT NULL,
    MainTel        VARCHAR2(15) NOT NULL
);

/*creating Production table*/
/*this table contains a primary key for PNum (production number)*/
CREATE TABLE Production (
    PNum           NUMBER(10) GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
    ProdTitle          VARCHAR2(255) NOT NULL,
    ProductionDirector VARCHAR2(255) NOT NULL,
    PlayAuthor     VARCHAR2(255) NOT NULL
);

```

```sql
/*creating Performance table*/
/* 'constraint' - Foreign keys, ties the necessary tables together */
/*this table contains a primary key for PerfNum (performance number)*/
/*also contains 2 foreign keys, PNum and TheatreNum which tie to the performance and production tables (called Performance_Production_FK and Performance_Theatre_FK)*/
CREATE TABLE Performance (
    PerfNum        NUMBER(10) GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
    PNum           NUMBER(10) NOT NULL,
    TheatreNum     NUMBER(10) NOT NULL,
    pDate          DATE NOT NULL,
    pHour          NUMBER(10) NOT NULL,
    pMinute        NUMBER(10) NOT NULL,
    Comments       VARCHAR2(4000),
    CONSTRAINT Performance_Production_FK FOREIGN KEY (PNum) REFERENCES Production (PNum),
    CONSTRAINT Performance_Theatre_FK FOREIGN KEY (TheatreNum) REFERENCES Theatre (TheatreNum)
);

/*creating Period table*/
/*this table contains a primary key for PeriodNum (period number)*/
CREATE TABLE Period (
    PeriodNum      NUMBER(10) GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
    Year           NUMBER(4) NOT NULL,
    Month          NUMBER(2) NOT NULL,
    Day            NUMBER(2) NOT NULL,
    Hour           NUMBER(2) NOT NULL,
    Minute         NUMBER(2) NOT NULL
);

/*creating the central fact table, TicketPurchase table*/
/*this table contains a primary key for PurchaseNum (purchase number)*/
/*contains foriegn keys to link to Client table (ClientNum), Performance table (PerfNum) and Period table (PeriodNum)*/
CREATE TABLE TicketPurchase (
    PurchaseNum    NUMBER(10) GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
    ClientNum      NUMBER(10),
    PerfNum        NUMBER(10),
    PeriodNum      NUMBER(10),
    PaymentMethod  VARCHAR2(50) NOT NULL,
    DeliveryMethod VARCHAR2(50) NOT NULL,
    TotalAmount    DECIMAL(10,2) NOT NULL,
    CONSTRAINT TicketPurchase_Client_FK FOREIGN KEY (ClientNum) REFERENCES Client (ClientNum),
    CONSTRAINT TicketPurchase_Performance_FK FOREIGN KEY (PerfNum) REFERENCES Performance (PerfNum),
    CONSTRAINT TicketPurchase_Period_FK FOREIGN KEY (PeriodNum) REFERENCES Period (PeriodNum)
);

```

```sql
/*inserting sample data*/
/*note how no primary keys are inserted, since they are automatically generated*/

/*inserting data into the Client table*/
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Mr.', 'John Doe', '123 Apple St.', '1234567890', 'john.doe@email.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Ms.', 'Jane Smith', '234 Banana Ave.', '0987654321', 'jane.smith@email.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Dr.', 'Alice Johnson', '345 Cherry Rd.', '1122334455', 'alice.johnson@email.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Mrs.', 'Emily White', '456 Date Blvd.', '2233445566', 'emily.white@email.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Mr.', 'Michael Brown', '567 Fig Lane.', '3344556677', 'michael.brown@email.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Mr.', 'Johnald Donaldson', '234 Gray St.', '912387543', 'johnaldsond@email.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Ms.', 'Jenifer Smithe', '224 Potate Ave.', '3248092', 'janefer.smith@email.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Dr.', 'Ryan Crate', '908 HUH Rd.', '9437202', 'alice@email.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Mrs.', 'Rupert', '1 the middle of nowhere st', '129873412', 'anon@hotmail.com');
INSERT INTO Client (ClientTitle, ClientName, Address, TelNo, Email) VALUES ('Mr.', 'Mick Miller', '120 Blackpool ln', '10284104', 'livecurrently@yahoo.com');

/*inserting data into the Theatre table*/
INSERT INTO Theatre (TheatreName, Address, MainTel) VALUES ('Grand Theatre', '123 Main St.', '1231231234');
INSERT INTO Theatre (TheatreName, Address, MainTel) VALUES ('Royal Theatre', '456 Grand Ave.', '3213214321');
INSERT INTO Theatre (TheatreName, Address, MainTel) VALUES ('Downtown Wigan Theatre', '789 Broadway St.', '2132132134');
INSERT INTO Theatre (TheatreName, Address, MainTel) VALUES ('Bolton Riverside Theatre', '135 River Rd.', '1321321324');
INSERT INTO Theatre (TheatreName, Address, MainTel) VALUES ('Teeside Theatre', '246 Hill St.', '2312312314');

/*inserting data into the Production table*/
INSERT INTO Production (ProdTitle, ProductionDirector, PlayAuthor) VALUES ('Romeo and Juliet and Mark', 'David Smith', 'Peter Shakespeare');
INSERT INTO Production (ProdTitle, ProductionDirector, PlayAuthor) VALUES ('Hamlet meets Porkchop', 'Sarah Johnson', 'Ryan Woolson');
INSERT INTO Production (ProdTitle, ProductionDirector, PlayAuthor) VALUES ('Macbeth with fries', 'Buck Williams', 'Frank Skinner');
INSERT INTO Production (ProdTitle, ProductionDirector, PlayAuthor) VALUES ('Othellololo', 'Mary Brown', 'Brent Peterson');
INSERT INTO Production (ProdTitle, ProductionDirector, PlayAuthor) VALUES ('King Lear 2', 'John Davis', 'Kyle');

/*inserting data into the Performance table*/
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (1, 1, TO_DATE('2023-11-10','YYYY-MM-DD'), 20, 0, 'Opening night!');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (1, 1, TO_DATE('2023-11-11','YYYY-MM-DD'), 20, 0, 'Second night special.');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (2, 1, TO_DATE('2023-11-12','YYYY-MM-DD'), 15, 30, 'Matinee.');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (2, 2, TO_DATE('2023-11-13','YYYY-MM-DD'), 20, 0, 'Evening show.');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (2, 2, TO_DATE('2023-11-14','YYYY-MM-DD'), 18, 0, 'Early evening show.');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (3, 2, TO_DATE('2023-11-15','YYYY-MM-DD'), 20, 0, 'Hooray.');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (3, 3, TO_DATE('2023-11-16','YYYY-MM-DD'), 20, 0, 'Woo Hoo!');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (4, 3, TO_DATE('2023-11-17','YYYY-MM-DD'), 15, 30, 'Strap in, ladys and gents!');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (4, 4, TO_DATE('2023-11-18','YYYY-MM-DD'), 20, 0, 'This is gonna be crazy!');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (5, 4, TO_DATE('2023-11-19','YYYY-MM-DD'), 18, 0, 'This one should be an alright show!');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (5, 5, TO_DATE('2023-11-20','YYYY-MM-DD'), 20, 0, 'This is gonna be even more crazy!');
INSERT INTO Performance (PNum, TheatreNum, pDate, pHour, pMinute, Comments) VALUES (5, 5, TO_DATE('2023-11-21','YYYY-MM-DD'), 18, 0, 'This one should also be an alright show!');
```

```sql
/*inserting data into the Period table*/
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2021, 8, 2, 19, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2021, 12, 8, 12, 45);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2021, 9, 15, 14, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2021, 10, 12, 15, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2021, 11, 19, 2, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2021, 12, 23, 17, 15);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2022, 1, 13, 9, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2022, 4, 8, 8, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2022, 12, 2, 3, 0);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2022, 5, 9, 7, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2022, 9, 11, 12, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2022, 11, 17, 14, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2023, 2, 19, 19, 15);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2023, 12, 11, 19, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2023, 12, 3, 21, 45);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2023, 6, 4, 17, 30);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2023, 12, 5, 18, 45);
INSERT INTO Period (Year, Month, Day, Hour, Minute) VALUES (2023, 12, 8, 3, 30);

/*inserting data into the TicketPurchase table*/
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (1,3,1,'Paypal', 'email', 25);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (2,12,2,'Credit Card', 'post', 30);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (2,9,3,'Cheque', 'email', 15);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (3,4,4,'Paypal', 'post', 15);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (4,4,5,'Cheque', 'email', 30);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (5,8,6,'Paypal', 'email', 45);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (5,2,7,'Credit Card', 'post', 15);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (5,5,8,'Paypal', 'email', 25);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (5,1,9,'Paypal', 'post', 20);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (6,11,10,'Cheque', 'email', 35);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (7,10,11,'Paypal', 'email', 50);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (8,7,12,'Cheque', 'email', 15);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (8,8,13,'Credit Card', 'post', 55);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (8,2,14,'Cheque', 'email', 10);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (8,1,15,'Paypal', 'post', 25);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (9,10,16,'Paypal', 'post', 30);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (9,4,17,'Credit Card', 'post', 10);
INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, PaymentMethod, DeliveryMethod, TotalAmount) VALUES (10,9,18,'Cheque', 'email', 45);
```

Screenshot of successful output

73	0.00	INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, P	1 row(s) inserted.	1
74	0.01	INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, P	1 row(s) inserted.	1
75	0.00	INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, P	1 row(s) inserted.	1
76	0.01	INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, P	1 row(s) inserted.	1
77	0.00	INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, P	1 row(s) inserted.	1
78	0.01	INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, P	1 row(s) inserted.	1
79	0.00	INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, P	1 row(s) inserted.	1
80	0.01	INSERT INTO TicketPurchase (ClientNum, PerfNum, PeriodNum, P	1 row(s) inserted.	1
81	0.04	SELECT TheatreName, Month, SUM(TotalAmount) "TotalSales" FR	21 rows selected.	21
82	0.02	SELECT ProdTitle, HOUR, COUNT(PurchaseNum) OVER (PARTITION	18 rows selected.	18
83	0.03	SELECT ProdTitle, Day, Email FROM Production JOIN Performa	2 rows selected.	2

Download

row(s) 1 - 83 of 83

83	83	0
Statements Processed	Successful	With Errors

gb_a743_sql_s21 gb_a743_sql_s21 en

Oracle APEX 22.2.1

Screenshots of analytical queries

```
--Shows the total sales of each theatre according to each month. Also shows total sales of each theatre
SELECT TheatreName, Month, SUM(TotalAmount) "TotalSales" --Selecting the Theatre name, month and summing up total sales
FROM Theatre JOIN Performance USING (TheatreNum)
JOIN TicketPurchase USING (PerfNum)
JOIN Period USING (PeriodNum) --Joining necessary tables using necessary keys
GROUP BY ROLLUP(TheatreName, Month); --groups the results by theatre name and month
```

Results Explain Describe Saved SQL History

THEATRENAME	MONTH	TotalSales
Grand Theatre	1	15
Grand Theatre	8	25
Grand Theatre	12	55
Grand Theatre	-	95
Royal Theatre	4	25
Royal Theatre	10	15
Royal Theatre	11	30
Royal Theatre	12	10
Royal Theatre	-	80
Teeside Theatre	5	35

More than 10 rows available. Increase rows selector to view more rows.

The above query is to extract the total sales of each theatre according to each month. The ROLLUP operator allows the output to show subtotals of each month and then a grand total for all time for each theatre, it also shows the total sales of all the theatres combined. This could be useful to MT since they could see which theatres are busy at certain times of the year. They could use this information to advertise accordingly.

```
--Shows the number of sales for each hour of the day for each production
SELECT ProdTitle, HOUR, COUNT(PurchaseNum) OVER (PARTITION BY HOUR) "NumberOfSales" --selecting the production name, hour and counting num of sales for each hour
FROM Production JOIN Performance USING (PNum)
JOIN TicketPurchase USING (PerfNum)
JOIN Period USING (PeriodNum); --Joining Necessary Tables
```

Results Explain Describe Saved SQL History

PRODTITLE	HOUR	NumberOfSales
Hamlet meets Porkchop	2	1
Othellololo	3	2
Romeo and Juliet and Mark	3	2
King Lear 2	7	1
Hamlet meets Porkchop	8	1
Romeo and Juliet and Mark	9	1
King Lear 2	12	2
King Lear 2	12	2
Macbeth with fries	14	2
Othellololo	14	2

More than 10 rows available. Increase rows selector to view more rows.

The above query is to extract the number of sales for each production for each hour of the day. The OVER PARTITION BY operator allows the query to select by each hour so that the data is organised. This could provide MT useful knowledge about what time people tend to buy theatre tickets, if certain customers tend to buy tickets late at night, then MT could start sending them advertisement emails in the evening for example. The period table's granularity allows for the hour to be extracted, even the minute can be extracted.

```
--Shows production name and customer emails of previous sales on a certain day
SELECT ProdTitle, Day, Email --selecting production title, day, email
FROM Production JOIN Performance USING (PNum)
JOIN TicketPurchase USING (PerfNum)
JOIN Client USING (ClientNum)
JOIN Period USING (PeriodNum) --joining necessary tables
WHERE Day = 2; --where the day is 2 (2nd of the month)
```

Results | Explain | Describe | Saved SQL | History

PRODTITLE	DAY	EMAIL
Hamlet meets Porkchop	2	john.doe@email.com
Romeo and Juliet and Mark	2	michael.brown@email.com

2 rows returned in 0.03 seconds Download

The above query is to extract the production name and customer email for each ticket purchased given a particular day of the month. In this example it is the second of the month. This could potentially be altered to be an extension of the above query, to extract the customers emails who buy at certain times of the day / month and email them advertisements accordingly.