Bike Store Dataset

Abstract:

The dataset consists of three interconnected tables: **customers, orders, and stores**, which track customer information, their orders, and the stores from where these orders originate. The dataset is designed to provide insights into the relationship between customers, their orders, and store locations. It can be used for customer analysis, order tracking, and store performance.

Summary:

- **customers table**: Contains personal details of customers, such as their first name, last name, phone number, and email address. It uniquely identifies each customer through customer_id.
- **orders table**: Tracks all the orders placed by customers, including the status of the order, important dates (such as order date, required shipping date, and shipped date), and the store_id where the order was placed. The customer_id serves as a foreign key linking to the customers table.
- **stores table**: Stores information about store locations, including the store's name, phone number, email, and its physical address (street, city, state, zip_code). The store_id links the stores to the orders table.

Table Overview:

1. **customers**:
    - **customer_id (int)**: Unique identifier for each customer.
    - **first_name (text)**: Customer's first name.
    - **last_name (text)**: Customer's last name.
    - **phone (text)**: Contact number of the customer.
    - **email (text)**: Email address of the customer.
2. **orders**:
    - **order_id (int)**: Unique identifier for each order.
    - **customer_id (int)**: References the customer_id from the customers table.
    - **order_status (int)**: Indicates the status of the order.
    - **order_date (text)**: Date when the order was placed.
    - **required_date (text)**: The required delivery date for the order.
    - **shipped_date (text)**: The date when the order was shipped.
    - **store_id (int)**: References the store where the order originated, linked to the stores table.
    - **staff_id (int)**: Represents the staff member handling the order.
3. **stores**:
    - **store_id (int)**: Unique identifier for each store.
    - **store_name (text)**: Name of the store.
    - **phone (text)**: Store's contact number.

- **email (text)**: Store’s email address.
- **street (text)**: Store's street address.
- **city (text)**: City where the store is located.
- **state (text)**: State of the store.
- **zip_code (int)**: ZIP code of the store.

Problem Statements

Que 1. How many orders have been made by Lizzette

Que 2. Display the details of orders where order_date, required_date are same

Que 3. Display order_id of orders where shipped_date is null

Que 4. List the names of customer having number

Que 5. Retrieve full name of customers who have placed order

Que 6. List top 5 orders where shipping was delayed

Que 7. Retrieve customer details who have placed more than 2 orders

Que 8. Get all orders placed in 2017

Que 9. select customers whose email domain is 'hotmail.com'

Que 10. Retrieve orders that were placed in the first quarter of 2018

Que 11. Retrieve first and last name whose first and last name contain no vowels except a and have been ordered more than 2 times

Que 12. Find the average time taken for shipping an order where shipping date was within 5 days from the order date

Que 13. Display all the customers who's first name starts with 's' and ends with 't' or who have lived in city of 'Jamaica' also avoid duplicate customer entries.

Que 14. Create a view of list of state

Que 15. Create a view that shows customers full name, city and most recent order date

Que 16. Count the occurence of each zip code in the customer tables which lies between 10000 and 11000.

Que 17. List all the customers who live in same city were store is located

Que 18. Count the number of orders supplied from each store

Que 19. Create a unique id for each customer by adding middle 2 number from zip_code and first 2 letter of first_name and last 2 letter of last_name

Que 20. Give the list of customers on basis of first_name and count of store of only those customers who have have placed orders in more than one store

Que 21. Give the list of customers on basis of first_name and count of store of only those customers who have have placed orders in more than one store

Que 22. Write a query to get all customers and their order details. Include customers even if they haven't placed any orders.

Que 23. Write a query to get every possible combination of customers and stores (even if they are not related). Include orders even if the customer information is missing

Que 24. Find the orders placed by customers from a specific list of cities ('Orchard Park', 'Fairport', 'Buffalo').

Que 25. Display the order date and categorize the orders based on the order date 'Old' for orders before 2016-05-01, 'Recent' for 2016-05-01 and onwards.

Que 26. Delete customers who haven't placed any orders.

Que 27. Update the city of customers who have placed an order in 2016-08-29 to 'Updated City' and display customer's id.

Que 1. How many orders have been made by Lizzette

Query

```
select customers.first_name, customers.last_name,
count(orders.customer_id) as Number_Of_Orders
from customers
join orders on customers.customer_id= orders.customer_id
where customers.first_name = "Lizzette"
group by customers.first_name,customers.last_name;
```

Output

First_name	Last_name	Number_Of_Orders
Lizzette	Stein	3

Que 2. Display the details of orders where order_date, required_date are same

Query

```
select * from orders
where order_date = required_date;
```

Or

```
select * from orders
where datediff(order_date, required_date)=0;
```

Output

order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id
70	50	3	11-02-2016	11-02-2016	NULL	3	9
76	9	3	16-02-2016	16-02-2016	NULL	2	6
158	117	3	04-04-2016	04-04-2016	NULL	3	9

185	24	3	20-04-2016	20-04-2016	NULL	1	2
239	13	3	27-05-2016	27-05-2016	NULL	3	9
246	17	3	30-05-2016	30-05-2016	NULL	2	6
264	5	3	10-06-2016	10-06-2016	NULL	1	3

Que 3. Display order_id of orders where shipped_date is null

Query

```
select order_id from orders
where shipped_date is null;
```

Output

order_id
70
76
158
185
239
246
264
349
350
357
391
407

Que 4. List the names of customer having number

Query

```
select first_name
from customers
where  phone is not null;
```

Output

first_name
Charolette
Latasha
Robby
Emmitt
Theo
Williemae

Janetta
Mozelle
Lolita
Tommie
Sarai

Que 5. Retrieve full name of customers who have placed order

Query

```
select concat(customers.first_name," ",customers.last_name) as full_name
from customers
inner  join orders
on customers.customer_id = orders.customer_id
where orders.order_id is not null;
```

Output

full_name
Johnathan Velazquez
Jaqueline Cummings
Joshua Robertson
Nova Hess
Arla Ellis
Sharyn Hopkins
Laureen Paul
Leslie Higgins
Neil Mccall
Alane Munoz
Tarra Guerrero
Marvin Mullins
Patience Clayton
Maribel William
Ellsworth Michael
Lea Key

Que 6. List top 5 orders where shipping was delayed

Query

```
select * from orders
where shipped_date > required_date
order by shipped_date - required_date desc
limit 5;
```

Output

order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id
4	175	4	03-01-2016	04-01-2016	05-01-2016	1	3
12	91	4	06-01-2016	08-01-2016	09-01-2016	1	2
14	258	4	09-01-2016	11-01-2016	12-01-2016	1	3
15	450	4	09-01-2016	10-01-2016	12-01-2016	2	7
20	923	4	14-01-2016	16-01-2016	17-01-2016	1	2

Que 7. Retrieve customer details who have placed more than 2 orders

Query

```
select customers.first_name, customers.last_name, count(orders.customer_id)
as total_orders
from customers
inner join orders
on customers.customer_id = orders.customer_id
group by customers.customer_id, customers.first_name, customers.last_name
having total_orders >2;
```

Output

first_name	last_name	total_orders
Cleotilde	Booth	3
Genoveva	Baldwin	3
Latasha	Hays	3
Robby	Sykes	3
Monika	Berg	3
Mozelle	Carter	3
Araceli	Golden	3
Williemae	Holloway	3
Corene	Wall	3
Lashawn	Ortiz	3
Caren	Stephens	3
Lorrie	Becker	3

Que 8. Get all orders placed in 2017

Query

```
select * from orders
where year(order_date)= 2017;
```

Output

order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id
636	532	4	03-01-2017	04-01-2017	05-01-2017	1	2
637	512	4	03-01-2017	05-01-2017	05-01-2017	2	6
638	1302	4	04-01-2017	05-01-2017	06-01-2017	2	7
639	550	4	06-01-2017	07-01-2017	07-01-2017	2	6
640	1093	4	06-01-2017	08-01-2017	07-01-2017	2	6
641	673	4	07-01-2017	10-01-2017	08-01-2017	1	3
642	847	4	07-01-2017	09-01-2017	10-01-2017	2	7
643	901	4	08-01-2017	09-01-2017	11-01-2017	1	3
644	439	4	08-01-2017	11-01-2017	10-01-2017	2	6
645	1199	4	08-01-2017	11-01-2017	10-01-2017	2	6
646	975	4	09-01-2017	12-01-2017	11-01-2017	1	2
647	283	4	09-01-2017	10-01-2017	11-01-2017	2	7
648	684	4	09-01-2017	11-01-2017	11-01-2017	2	6
649	1153	4	10-01-2017	11-01-2017	13-01-2017	1	3

Que 9. select customers whose email domain is 'hotmail.com'

Query

```
select * from customers

where email like '%hotmail.com';
```

Output

cust_id	first_name	last_name	phone	email	street	city	state	zip_code
6	Lyndsey	Bean	NULL	lyndsey.bean@hotmail.com	769 West Road	Fairport	NY	14450
7	Latasha	Hays	(716) 986-3359	latasha.hays@hotmail.com	7014 Manor Station Rd.	Buffalo	NY	14215
12	Robby	Sykes	(516) 583-7761	robby.sykes@hotmail.com	486 Rock Maple Street	Hempstead	NY	11550
14	Garry	Espinoza	NULL	garry.espinoza@hotmail.com	7858 Rockaway Court	Forney	TX	75126
16	Emmitt	Sanchez	(212) 945-8823	emmitt.sanchez@hotmail.com	461 Squaw Creek Road	New York	NY	10002
28	Jeanice	Frost	NULL	jeanice.frost@hotmail.com	76 Devon Lane	Ossining	NY	10562
33	Deloris	Burke	NULL	deloris.burke@hotmail.com	895 Edgemont Drive	Palos Verdes Peninsula	CA	90274
36	Bernita	Mcdaniel	NULL	bernita.mcdaniel@hotmail.com	2 Peg Shop Ave.	Liverpool	NY	13090
47	Bridgette	Guerra	NULL	bridgette.guerra@hotmail.com	9982 Manor Drive	San Lorenzo	CA	94580
52	Tu	Ramirez	NULL	tu.ramirez@hotmail.com	24 W. Courtland Street	East Elmhurst	NY	11369
54	Fran	Yang	NULL	fran.yang@hotmail.com	440 Pearl St.	Utica	NY	13501
55	Diana	Guerra	NULL	diana.guerra@hotmail.com	45 Chapel Ave.	Merrick	NY	11566
56	Lolita	Mosley	(281) 363-3309	lolita.mosley@hotmail.com	376 S. High Ridge St.	Houston	TX	77016
62	Alica	Hunter	NULL	alica.hunter@hotmail.com	8 San Juan Drive	East Elmhurst	NY	11369

Que 10. Retrieve orders that were placed in the first quarter of 2018

Query

```
select * from orders

where order_date

between '2018-01-01' and '2018-03-31';
```

Output

order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id
1324	862	4	01-01-2018	04-01-2018	03-01-2018	2	6
1325	68	4	01-01-2018	02-01-2018	03-01-2018	3	8
1326	567	4	01-01-2018	03-01-2018	02-01-2018	3	9
1327	1026	4	02-01-2018	03-01-2018	05-01-2018	2	6
1328	1083	4	02-01-2018	03-01-2018	03-01-2018	2	6
1329	443	4	04-01-2018	05-01-2018	06-01-2018	1	3
1330	761	4	04-01-2018	05-01-2018	07-01-2018	2	7
1331	1122	4	05-01-2018	06-01-2018	08-01-2018	2	6
1332	256	4	06-01-2018	07-01-2018	09-01-2018	2	7
1333	203	4	06-01-2018	07-01-2018	07-01-2018	3	8
1334	425	4	07-01-2018	09-01-2018	10-01-2018	2	6
1335	955	4	07-01-2018	08-01-2018	08-01-2018	2	7
1336	904	4	09-01-2018	10-01-2018	12-01-2018	1	2
1337	970	4	09-01-2018	12-01-2018	12-01-2018	2	7

Que 11. Retrieve first and last name whose first and last name contain no vowels except a vowel and have been ordered more than 2 times

Query

```
select customers.first_name,customers.last_name, count(orders.customer_id) as order_count from customers

inner join orders on customers.customer_id = orders.customer_id

where first_name not like "%e%" and last_name not like "%e%"

and first_name not like "%i%" and last_name not like "%i%"

and first_name not like "%o%" and last_name not like "%o%"

and first_name not like "%u%" and last_name not like "%u%"

group by customers.first_name, customers.last_name

having order_count >2 ;
```

Output

first_name	last_name	order_count
Latasha	Hays	3

Que 12. Find the average time taken for shipping an order where shipping date was within 5 days from the order date

Query

```
select avg(datediff(shipped_date,order_date)) as Average from orders

where datediff(shipped_date, order_date) <= 5;
```

Output

Average
1.982

Que 13. Display all the customers who's first name starts with 's' and ends with 't' or who have lived in city of 'Jamaica' also avoid duplicate customer entries.

Query

```
select distinct * from customers

where first_name like 's%t'

or  city= 'Jamaica';
```

Output

cust_id	first_name	last_name	phone	email	street	city	state	zip_code
223	Scarlet	Yates	NULL	scarlet.yates@hotmail.com	687 Shirley St.	Patchogue	NY	11772
335	Scarlet	Reed	NULL	scarlet.reed@hotmail.com	151 S. Myers Court	Bronx	NY	10451
205	Harris	Pittman	NULL	harris.pittman@yahoo.com	404 Cherry Hill Road	Jamaica	NY	11432
272	Deja	Chaney	NULL	deja.chaney@hotmail.com	52 Woodside Ave.	Jamaica	NY	11432
352	Josef	Greer	NULL	josef.greer@gmail.com	92 Summer Court	Jamaica	NY	11432
439	Sunshine	Rosario	NULL	sunshine.rosario@aol.com	1 Chestnut Lane	Jamaica	NY	11432
456	Serina	Hensley	NULL	serina.hensley@yahoo.com	53 Proctor Drive	Jamaica	NY	11432
505	Wilda	Petersen	NULL	wilda.petersen@aol.com	899 Helen St.	Jamaica	NY	11432
575	Laurence	Christian	NULL	laurence.christian@gmail.com	2 Maple St.	Jamaica	NY	11432
620	Eric	Hardin	NULL	eric.hardin@gmail.com	39 S. Parker Drive	Jamaica	NY	11432
1086	Cori	Schwartz	NULL	cori.schwartz@msn.com	43 Lancaster Lane	Jamaica	NY	11432
1375	Carlena	Salinas	NULL	carlena.salinas@gmail.com	1 Wagon Street	Jamaica	NY	11432
1393	Vivian	Deleon	NULL	vivian.deleon@msn.com	9925 Hamilton Street	Jamaica	NY	11432

Que 14. Create a view of list of state

Query

```
create view unique_state_view as

select distinct state from customers;
```

select * from unique_state_view;

Output

state
NY
CA
TX

Que 15. Create a view that shows customers full name, city and most recent order date

Query

```
create view recent_orders_of_customer as
select concat(customers.first_name," ",customers.last_name) as Fullname,customers.city,
max(orders.order_date) as recent_order
from customers
join orders on customers.customer_id= orders.customer_id
group by fullname, city;
select * from recent_orders_of_customer;
```

Output

Fullname	city	recent_order
Johnathan Velazquez	Pleasanton	01-01-2016
Jaqueline Cummings	Huntington Station	01-01-2016
Joshua Robertson	Patchogue	02-01-2016
Nova Hess	Duarte	11-04-2018
Arla Ellis	Utica	03-01-2016
Sharyn Hopkins	Baldwinsville	01-04-2018
Laureen Paul	Bellmore	04-01-2016
Leslie Higgins	Saratoga Springs	04-01-2016
Neil Mccall	San Carlos	21-04-2018
Alane Munoz	Yonkers	05-01-2016
Tarra Guerrero	Auburn	05-01-2016
Marvin Mullins	San Diego	17-04-2018
Patience Clayton	Niagara Falls	08-01-2016
Maribel William	Torrance	09-01-2016

Que 16. Count the occurence of each zip code in the customer tables which lies between 10000 and 11000.

Query

```
select zip_code, count(zip_code) from customers
where zip_code between 10000 and 11000
```

group by zip_code

order by count(zip_code) desc;

Output

zip_code	count(zip_code)
10550	20
10583	17
10562	13
10950	11
10952	11
10977	10
10451	9
10573	8
10002	7
10541	7
10301	7
10512	7
10801	7
10701	6
10956	5
10598	5
10954	4

Que 17. List all the customers who live in same city were store is located

Query

select concat(customers.first_name," ", customers.last_name) as customer_name,

customers.city, stores.store_name

from customers join stores

on customers.city = stores.city;

Output

customer_name	city	store_name
Jayne Kirkland	Rowlett	Rowlett Bikes
Ashanti Parks	Baldwin	Baldwin Bikes
Omer Estrada	Rowlett	Rowlett Bikes
Dorine Roberson	Santa Cruz	Santa Cruz Bikes
Javier Nichols	Rowlett	Rowlett Bikes
Justina Jenkins	Baldwin	Baldwin Bikes
Ginette Edwards	Baldwin	Baldwin Bikes
Jennette Wooten	Rowlett	Rowlett Bikes
Lashawna Richardson	Baldwin	Baldwin Bikes
Christopher Richardson	Santa Cruz	Santa Cruz Bikes

Cyndi Dyer	Baldwin	Baldwin Bikes
Sam Lester	Baldwin	Baldwin Bikes
Kecia Olsen	Santa Cruz	Santa Cruz Bikes
Vallie Dixon	Rowlett	Rowlett Bikes
Stephanie Browning	Rowlett	Rowlett Bikes
Trinidad Mcclain	Baldwin	Baldwin Bikes
Ping Quinn	Rowlett	Rowlett Bikes
Patsy Russo	Rowlett	Rowlett Bikes
Harold O'connor	Santa Cruz	Santa Cruz Bikes
Carola Johns	Santa Cruz	Santa Cruz Bikes

Que 18. Count the number of orders supplied from each store

Query

```
select stores.store_id, stores.store_name , count(orders.store_id) as count_of_order
from  orders inner join stores on orders.store_id= stores.store_id
group by stores.store_id, stores.store_name;
```

Output

store_id	store_name	count_of_order
1	Santa Cruz Bikes	348
2	Baldwin Bikes	1093
3	Rowlett Bikes	174

Que 19. Create a unique id for each customer by adding middle 2 number from zip_code and first 2 letter of first_name and last 2 letter of last_name

Query

```
select first_name, last_name,
concat(substr(zip_code,(length(zip_code)/2),2),
left(first_name,2),
right(last_name,2))as Unique_id
from customers
group by first_name,last_name,Unique_id;
```

Output

first_name	last_name	Unique_id
Debra	Burks	12Deks
Kasha	Todd	00Kadd
Tameka	Fisher	27Taer
Daryl	Spence	55Dace

Charolette	Rice	82Chce
Lyndsey	Bean	45Lyan
Latasha	Hays	21Lays
Jacquline	Duncan	37Jaan
Genoveva	Baldwin	05Gein
Pamelia	Newman	95Paan
Deshawn	Mendoza	95Deza
Robby	Sykes	55Roes
Lashawn	Ortiz	60Laiz
Garry	Espinoza	12Gaza

Que 20. Give the list of customers on basis of first_name and count of store of only those customers who have have placed orders in more than one store

Query

```
select customers.first_name as Customer, count(distinct orders.store_id) as Number_of_Store
from customers
join orders join stores
on customers.customer_id = orders.customer_id
and stores.store_id= orders.store_id
group by customer
having  count(distinct orders.store_id)>1;
```

Output

Customer	Number_of_Store
Adam	2
Adriene	2
Agatha	2
Alane	2
Alejandro	2
Aleta	2
Angele	2
Arvilla	2
Bernetta	2
Bettie	2
Carola	2
Cesar	2
Chelsey	2
Christel	2

Que 21. Write a query to get all customers and their order details. Include customers even if they haven't placed any orders.

Query

```
select customers.first_name, customers.last_name, orders.order_date
```

from customers left join orders

on customers.customer_id= orders.customer_id;

Output

first_name	last_name	order_date
Debra	Burks	18-11-2018
Debra	Burks	18-04-2018
Debra	Burks	09-12-2016
Kasha	Todd	09-04-2018
Kasha	Todd	21-08-2017
Kasha	Todd	05-02-2017
Tameka	Fisher	21-10-2018
Tameka	Fisher	06-04-2018
Tameka	Fisher	27-03-2018
Daryl	Spence	18-04-2018
Daryl	Spence	21-11-2017
Daryl	Spence	07-02-2017
Charolette	Rice	17-04-2018
Charolette	Rice	24-11-2016
Charolette	Rice	10-06-2016
Lyndsey	Bean	06-09-2018
Lyndsey	Bean	27-04-2018
Lyndsey	Bean	14-08-2017
Latasha	Hays	17-06-2018

Que 22. Write a query to get all order details and their corresponding customer information. Include orders even if the customer information is missing

Query

select orders.order_id, orders.order_date, customers.customer_id, customers.first_name, customers.last_name

from orders right join customers

on orders.customer_id = customers.customer_id;

Output

order_id	order_date	customer_id	first_name	last_name
1613	18-11-2018	1	Debra	Burks
1555	18-04-2018	1	Debra	Burks
599	09-12-2016	1	Debra	Burks
1509	09-04-2018	2	Kasha	Todd
1084	21-08-2017	2	Kasha	Todd
692	05-02-2017	2	Kasha	Todd
1612	21-10-2018	3	Tameka	Fisher
1496	06-04-2018	3	Tameka	Fisher
1468	27-03-2018	3	Tameka	Fisher
1556	18-04-2018	4	Daryl	Spence
1259	21-11-2017	4	Daryl	Spence

700	07-02-2017	4	Daryl	Spence
1544	17-04-2018	5	Charolette	Rice
571	24-11-2016	5	Charolette	Rice
264	10-06-2016	5	Charolette	Rice
1611	06-09-2018	6	Lyndsey	Bean
1592	27-04-2018	6	Lyndsey	Bean
1059	14-08-2017	6	Lyndsey	Bean
1604	17-06-2018	7	Latasha	Hays

Que 23. Write a query to get every possible combination of customers and stores (even if they are not related).

Query

```
select customers.customer_id , stores.store_id

from customers cross join stores

order by customers.customer_id desc;
```

Output

customer_id	store_id
1445	1
1445	3
1445	2
1444	1
1444	3
1444	2
1443	1
1443	3
1443	2
1442	1
1442	3
1442	2
1441	1
1441	3
1441	2
1440	1
1440	3
1440	2
1439	1

Que 24. Find the orders placed by customers from a specific list of cities ('Orchard Park', 'Fairport', 'Buffalo').

Query

```
SELECT order_id, customer_id, order_date

FROM orders

WHERE customer_id IN (

   SELECT customer_id
```

```
    FROM customers
    WHERE city IN ('Orchard Park', 'Fairport', 'Buffalo')
);
```

Output

order_id	customer_id	order_date
29	437	20-01-2016
34	80	22-01-2016
104	7	03-03-2016
192	728	27-04-2016
220	1358	12-05-2016
265	682	10-06-2016
375	536	17-08-2016
386	1362	20-08-2016
469	1089	29-09-2016
516	111	21-10-2016
599	1	09-12-2016
621	1157	23-12-2016
724	864	19-02-2017
739	49	28-02-2017
814	629	02-04-2017
887	1195	13-05-2017
978	1123	26-06-2017
1044	1270	07-08-2017
1059	6	14-08-2017

Que 25. Display the order date and categorize the orders based on the order date 'Old' for orders before 2016-05-01, 'Recent' for 2016-05-01 and onwards.

Query

```
SELECT order_id, order_date,
CASE
    WHEN order_date < '2016-05-01' THEN 'Old'
    ELSE 'Recent'
END AS order_category
from orders;
```

Output

order_id	order_date	order_category
1	01-01-2016	Old
2	01-01-2016	Old
3	02-01-2016	Old

4	03-01-2016	Old
5	03-01-2016	Old
6	04-01-2016	Old
7	04-01-2016	Old
8	04-01-2016	Old
9	05-01-2016	Old
10	05-01-2016	Old
11	05-01-2016	Old
12	06-01-2016	Old
13	08-01-2016	Old
14	09-01-2016	Old
15	09-01-2016	Old
16	12-01-2016	Old
17	12-01-2016	Old
18	14-01-2016	Old
19	14-01-2016	Old

Que 26. Delete customers who haven't placed any orders.

Query

```
delete from customers
where customer_id not in( select customer_id from orders);
```

Que 27. Update the city of customers who have placed an order in 2016-08-29 to 'Updated City' and display customer's id.

Query

```
UPDATE customers
SET city = 'Updated City'
WHERE customer_id IN (
   SELECT customer_id
   FROM orders
   WHERE order_date = '2016-08-29');
select customer_id from customers
where city ='Updated City';
```

Output

customer_id
447
1001
1295