



Symbiosis Institute of Technology
A DBMS Project Report on
ADIDAS DATABASE MANAGEMENT SYSTEM

Submitted by

Animesh Jain

(18070124009)

Vaishnavi Deshpande

(18070124021)

Pragya Mandloi

(18070124048)

Under the Guidance of

(Prof. Shruti Patil)

Department of Computer Science

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Index

1. Introduction-----	1
2. Problem Statement-----	2
3. System Architecture-----	3
4. Functional Requirements-----	4
5. Entities, Attributes, Keys, Relationships-----	5
6. E-R diagram-----	9
7. Relational schema-----	10
8. Codd's Rule-----	14
9. Functional dependencies of each relation-----	21
10. Kind of anomalies in relational schema-----	25
11. Normalization of relational schema-----	25
12. Database Implementation-----	27
13. Query Execution-----	35
14. Functions-----	42
15. Procedures-----	46
16. Triggers-----	53

Introduction:

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system. We aim to depict a prototype of the database of Adidas in India. It includes the different categories of products that a customer can browse through. Hence, we plan on showing the way how adidas manages such a diverse database. The implementation of this project will be in MySQL.

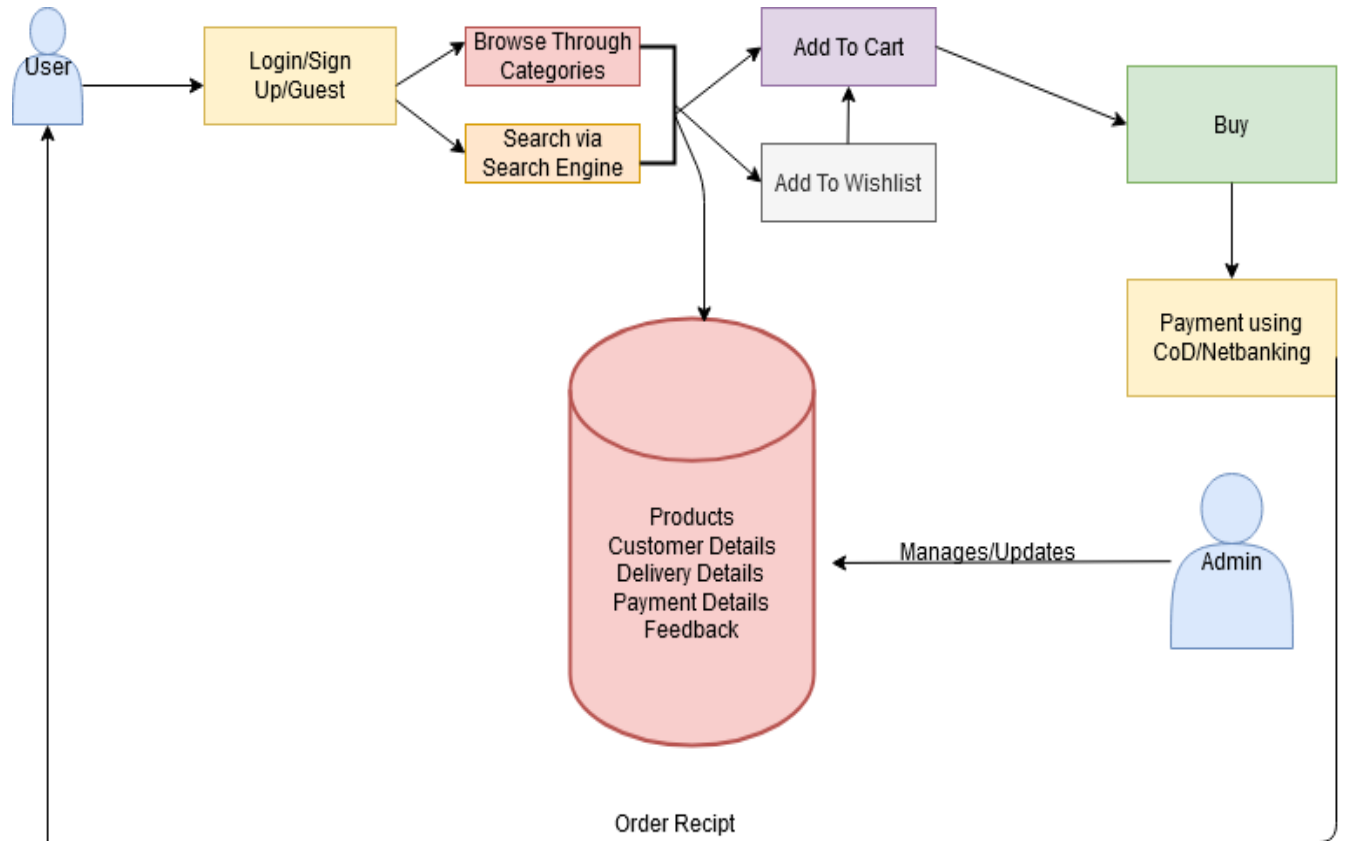


Problem Statement:

Adidas is a multinational brand which centralizes itself around sports related products. A database as exhaustive as Adidas' needs a proper database management system to function in a feasible and fluent way. We have tried our hand at making this database manageable and easy to work with.

Hence, we have tried to cover as many factors as possible to make it realistic and to replicate the official shopping website of the brand.

SYSTEM ARCHITECTURE:



FUNCTIONAL REQUIREMENTS:

1)Customer Module:

- Login/Sign up:
A customer can login/sign up as a guest/member. Input- Username, Password, Contact no.
- Browsing:
A customer chooses and browses through products of his/her choice which belong to respective categories.
- Cart Module:
A customer can add the products to his/her shopping cart which will be associated to the customer's account.
- Wishlist:
A customer can add products to the wishlist to review /buy later. The products from the wishlist can be added to the cart and vice versa according to the customer's requirement.
- Order Checkout:
The total amount of the order is displayed, which has to be paid by the customer. The order status and its details are also displayed to the customer.
- Payment:
Customer can pay using different modes of payment, viz. Card, Net Banking, Wallet, Cash on Delivery.
- Feedback/Review:
Customer can review and give feedback.

2)Adidas Team Module:

- Updates the new products and their respective sections.
- Manages the importation and manufacture of products.
- Keeps an account of all the transactions and purchases.
- Provides customer service to the users.
- Launches seasonal and special products.

Identifying Entities, Attributes and Relationships:

- Adidas Team:

Team_ID, Department

Primary Key: Team_ID

- (Adidas_Team) *has* (Employee)
- (Adidas_Team) *provides* (Customer_Service)
- (Adidas_Team) *manages* (Product)
- (Adidas_Team) *accepts* (Feedback/review)
- (Adidas_Team) *verifies* (Payment)

- Product:

Product ID, actual cost, offer, final amount, name, tag, color, stock avail, size, material, specifications, warrantee, country of origin

Primary Key: Product ID

- (Product) *managed_by* (Adidas Team)
- (Product) *have* (Collections)
- (Product) *is_of* (Brands)
- (Product) *moves_to* (Shopping Cart)
- (Product) *belongs_to* (Categories)
- (Product) *belongs_to* (Sport)
- (Product) *belongs_to* (Section)

- Country_of_Origin:

Country_ID, Country_Name

- (Country_of_origin) *manufactures* (product)

- Collections:

Collection_id, name

Primary Key: Collection_ID

- (Collections) *have* (Product)

- Sport:

Type_ID, name

Primary Key: Type_ID

- (Sport) *has* (Section)

- (Sport) *has* (Category)
- (Sport) *contains* (Product)

- **Brands:**
 Brand_ID, name
Primary Key: Brand_ID
 - (Brands) *have* (Product)

- **Category:**
 Category_ID, name
Primary Key: Category_ID
 - (Category) *has* (Section)
 - (Category) *has* (Sport)
 - (Category) *contains* (Product)

- **Section:**
 Section_ID, name
Primary Key: Section_ID
 - (Section) *has* (Category)
 - (Section) *has* (Sport)
 - (Section) *contains* (Product)

- **Feedback/Review:**
 No_of_stars, Rev_ID, Customer_ID, Team_ID
Primary Key: Rev_ID

- **Customer:**
 Customer_ID, name- first name, last name, contact no, address- pincode, street, city, state, landmark
Primary Key: Customer_ID
 - (Customer) *logs_in/Signs_up* (Account)
 - (Customer) *provides* (Feedback/Review)
 - (Customer) *contacts* (Customer_service)

- **Account:**
 Account_ID, Email_ID, Password
Primary Key: Account_ID
 - (Account) *views/selects* (Product)
 - (Account) *has* (Shopping_cart)
 - (Account) *has* (Coupon)

- (Account) *has* (Wishlist)
- (Account) **ISA**(Guest)
- (Account) **ISA**(Register)

- Customer Service:
Email_ID, Contact_no., Cs_ID
Primary Key: Cs_ID

- Coupon:
Coupon_No., coupon_code, amount, validity
Primary Key: Coupon_no.
(Coupon) *applied_on* (Order)

- Shopping Cart:
Cart_ID
Primary Key: Cart_ID
 - (Shopping cart) *checkout* (Order)
 - (Shopping cart) *move_to* (Wishlist)

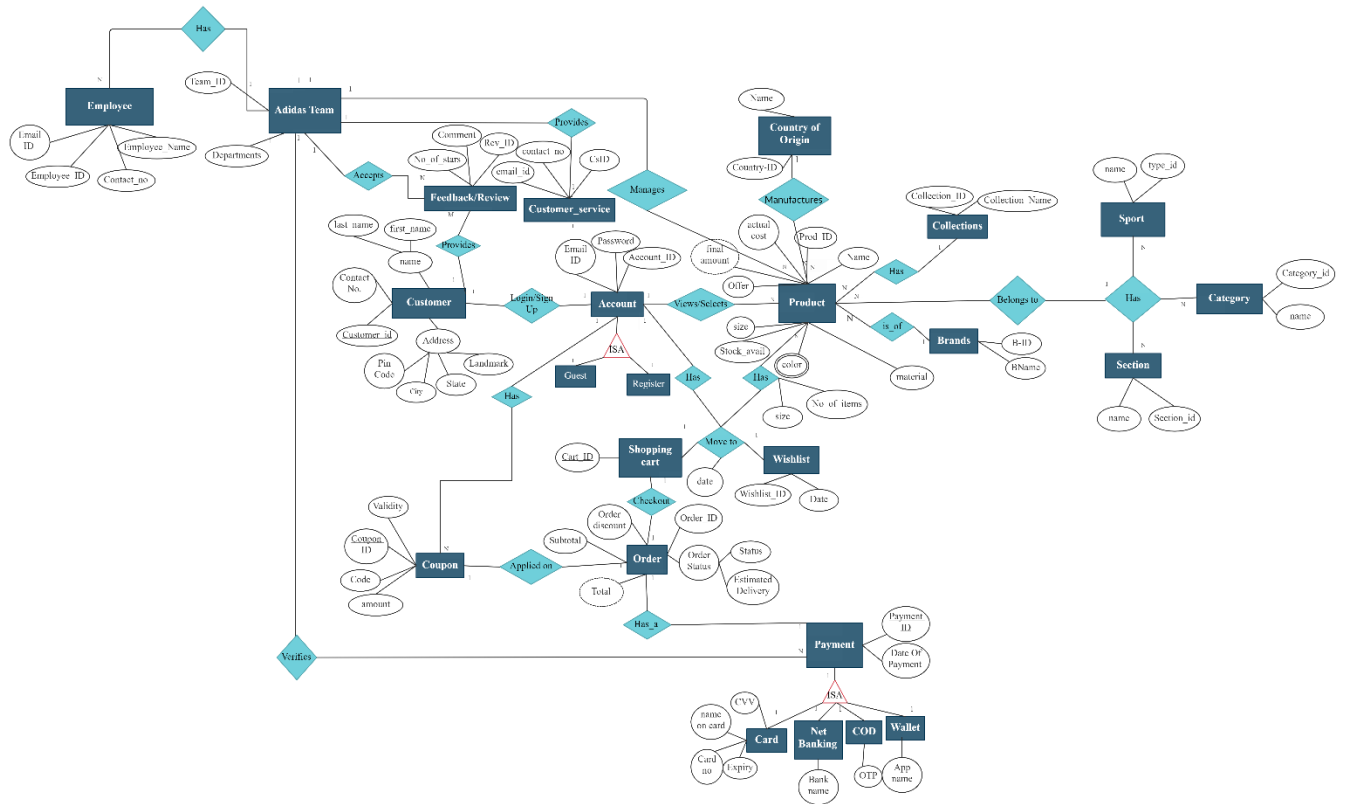
- Wishlist:
Wishlist_ID, date
Primary Key: Wishlist_ID
 - (Wishlist) *move_to* (Shopping cart)

- Order:
Order_ID, order discount, subtotal total, offer, order status- status, estimated delivery
Primary Key: Order_ID
 - (Order) *has_a* (Payment)

- Payment:
Payment_ID, date_of_payment
Primary Key: Payment_ID
 - (Payment) **ISA** (card)
 - (Payment) **ISA** (COD)
 - (Payment) **ISA** (wallet)
 - (Payment) **ISA** (netbanking)

- Card:
CVV, Name_on_card, Expiry, Card_no, Payment_ID, date_of_payment
- COD:
OTP, Payment_ID
- Wallet:
App_name, Payment_ID
- Netbanking:
Bank_name, Payment_ID

ER DIAGRAM:



Link to ER diagram

RELATIONAL SCHEMA:

Adidas team:

Team_ID	Departments
----------------	-------------

Employee:

Employee_ID	Email_ID	Employee_name	Contact_no.	<u>Team_ID</u>
--------------------	----------	---------------	-------------	----------------

Customer:

Customer_ID	First_name	Last_name	Contact_no.	Pin_code	Landmark	<u>Account_ID</u>
--------------------	------------	-----------	-------------	----------	----------	-------------------

Account:

Account_ID	Email_ID
-------------------	----------

Guest:

Account_ID

Registered:

Account_ID	Password
-------------------	----------

Product:

P_ID	Name	Cost	Offer	material	B_ID	Team_ID	Collection_ID	Country_ID
-------------	------	------	-------	----------	------	---------	---------------	------------

Color:

P_ID	Color
-------------	--------------

Country_of_origin:

Country_ID	Country_Name
-------------------	--------------

Collection:

Collection_ID	Name
----------------------	------

Sport:

Type_ID	Name
----------------	------

Brand:

Brand_ID	Name
-----------------	------

Category:

Category_ID	Name
--------------------	------

Section:

Section_ID	Name
-------------------	------

Feedback/Review:

Rev_ID	No_of_stars	<u>Team_ID</u>	<u>Customer_ID</u>
---------------	-------------	----------------	--------------------

Customer Service:

Cs_ID	Email_ID	Contact_no.	<u>Team_ID</u>
--------------	----------	-------------	----------------

Wishlist:

Wishlist_id	Date
--------------------	------

Cart:

Cart_id

Account_has_cart:

Account_ID	Wishlist_ID	Cart_ID
-------------------	--------------------	----------------

Cart_has_product:

L_ID	Cart_ID	P_ID	Wishlist_ID	No_of_items	Size
-------------	----------------	-------------	--------------------	--------------------	-------------

Coupon:

Coupon_ID	coupon_code	amount	validity	<u>Account_ID</u>
------------------	-------------	--------	----------	-------------------

Order:

Order_id	discount	subtotal	status	Estimated_delivery	<u>Cart_id</u>	<u>Coupon_id</u>
-----------------	----------	----------	--------	--------------------	----------------	------------------

Payment:

Payment_id	Date_of_payment	<u>Order_id</u>	<u>Team_id</u>
-------------------	-----------------	-----------------	----------------

Card:

Payment_id	CVV	Name_on_card	Expiry	Card_no
-------------------	-----	--------------	--------	---------

COD:

Payment_id	OTP
-------------------	-----

Wallet:

Payment_id	App_name
-------------------	----------

Netbanking:

Payment_id	Bank_name
-------------------	-----------

Codd's Rule

Rule 1: Information Rule

The data stored in a database, may it be user data or metadata, must be a value of some table cell. Everything in a database must be stored in a table format.

→ **Yes**, this rule is applicable to our project since all the data in our project will be represented and stored in tabular form.

Rule 2: Guaranteed Access Rule

Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value). No other means, such as pointers, can be used to access data.

→ **Yes**, this rule is applicable to our project as every single data element (value) is guaranteed to be accessible logically with a combination of primary key and various elements.

Rule 3: Systematic Treatment of NULL Values

The NULL values in a database must be given a systematic and uniform treatment. This is a very important rule because a NULL can be interpreted as one the following – data is missing, data is not known, or data is not applicable.

→ **Yes**, this rule is applicable to our project because all the NULL values have been given an uniform treatment.

Rule 4: Active Online Catalog

The structure description of the entire database must be stored in an online catalog, known as **data dictionary**, which can be accessed by authorized users. Users can use the same query language to access the catalog which they use to access the database itself.

→ **Yes**, this rule is applicable to our project because the admin and users can access the database and catalogue with same query language.

Rule 5: Comprehensive Data Sub-Language Rule

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation.

Example: SQL, etc. If the database allows access to the data without the use of this language, then that is a violation.

➔ **Yes**, this rule is applicable to our project.

Rule 6: View Updating Rule

All the views of a database, which can theoretically be updated, must also be updatable by the system.

➔ **Yes**, this rule is applicable to our project as all the views can be updated by the system.

Rule 7: High-Level Insert, Update, and Delete Rule

A database must support high-level insertion, updation and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

➔ **Yes**, this rule is applicable to our project

Rule 8: Physical Data Independence

The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

➔ **Yes**, this rule is applicable to our project as data is not affected by the change caused by the access of different applications.

Example: If some file supporting the table is renamed or removed from one disk to another, it should not affect the application.

Rule 9: Logical Data Independence

Any change in logical data must not affect the applications using it.

➔ **Yes**, this rule is applicable to our project

Example: If two tables are merged or one is split into two different tables, there should be no impact or change on the user application

Rule 10: Integrity Independence

A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

➔ **Yes**, this rule is applicable to our project as it is independent of the platform and application.

Rule 11: Distribution Independence

The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

➔ **Yes**, this rule is applicable to our project as the user won't be able to see the distribution of data over various locations.

Rule 12: Non-Subversion Rule

If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.

➔ **No**, this rule is not applicable to our project.

Dependency

Adidas team:

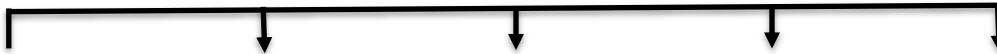
Full dependency



Team_ID	Department_name
----------------	-----------------

Employee:

Full dependency



Employee_ID	Email_ID	Employee_name	Contact_no.	<u>Team_ID</u>
--------------------	----------	---------------	-------------	----------------

Customer:

Full dependency



Customer_ID	First_name	Last_name	Contact_no	Pin_code	City	State	Landmark	<u>Account_ID</u>
--------------------	------------	-----------	------------	----------	------	-------	----------	-------------------



Transitive dependency

Account:



Account_ID	Password
-------------------	----------

Registered



Account_ID	Password
-------------------	----------

Guest:

No functional dependency

Account_ID

Product:



P_ID	Name	Cost	Offer	material	B_ID	Team_ID	Collection_ID	Country_ID
-------------	------	------	-------	----------	------	---------	---------------	------------

Color:



<u>P_ID</u>	Color
--------------------	-------

Collection:

Full dependency



Collection_ID	Name
----------------------	------

Belongs_to:

No dependency

P_ID	Category_ID	Section_ID	Sport_ID
-------------	--------------------	-------------------	-----------------

Has:

No dependency

<u>Category_ID</u>	<u>Section_ID</u>	<u>Sport_ID</u>
--------------------	-------------------	-----------------

Sport:

Full dependency



Sport_ID	Name
-----------------	------

Brand:

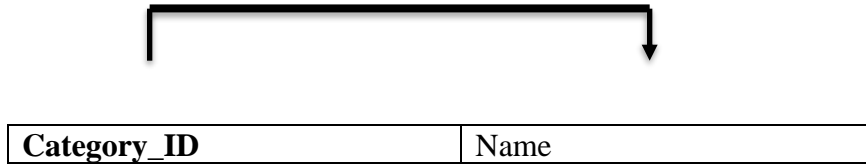
Full dependency



B_ID	Name
-------------	------

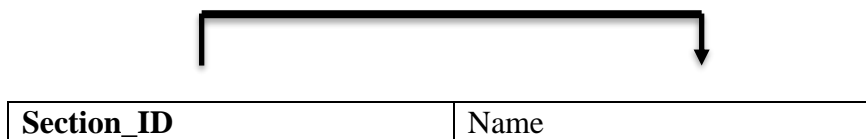
Category:

Full dependency



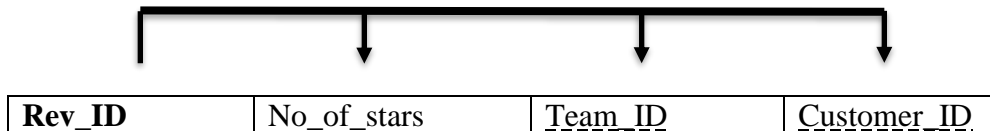
Section:

Full dependency



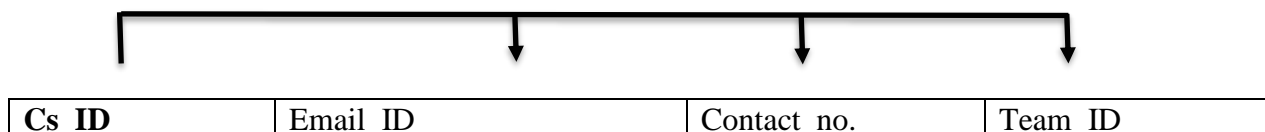
Feedback/Review:

Full dependency



Customer Service:

Full dependency



Country of Origin:

Full dependency



Country_ID	Country_Name
------------	--------------

Wishlist:

Full dependency



Wishlist_ID	date
-------------	------

Cart:

No dependency

Cart_id

Account_Has_Cart:

No dependency

Acc_ID	Wishlist_ID	Cart_ID
--------	-------------	---------

Cart_Has_Product:

Full dependency



L_ID	Cart_ID	P_ID	Wishlist_ID	No_of_items	Size
-------------	----------------	-------------	--------------------	-------------	------

Coupon:

Full dependency



Coupon_id	coupon_code	Amount	validity	<u>Account_ID</u>
------------------	-------------	--------	----------	-------------------

Order:

Full dependency



Order_id	Discount	Subtotal	Status	Estimated_delivery	<u>Cart_id</u>	<u>Coupon_id</u>
-----------------	----------	----------	--------	--------------------	----------------	------------------

Payment:

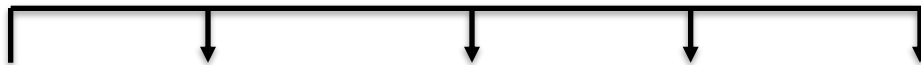
Full dependency



Payment_id	Date_of_payment	<u>Order_id</u>	<u>Team_id</u>
------------	-----------------	-----------------	----------------

Card:

Full dependency



Payment_id	CVV	Name_on_card	Expiry	Card_no
------------	-----	--------------	--------	---------

COD:

Full dependency



Payment_id	OTP
------------	-----

Wallet:

Full dependency



Payment_id	App_name
------------	----------

Netbanking:

Full dependency



Payment_id	Bank_name
-------------------	------------------

Anomalies

There are Insertion, Deletion & Updation anomalies present in 'Customer Table'.

Normalization

All the tables are in normalized form, 3NF, apart from the customer table which is in 2NF.

Full dependency

Customer_ ID	First_na me	Last_na me	Contact_ no	Pin_co de	Cit y	Stat e	Landma rk	<u>Account_I D</u>
-------------------------	----------------	---------------	----------------	--------------	----------	-----------	--------------	------------------------

2NF

Normalization:

Customer_ID	First_name	Last_name	Contact_no	Pin_code	Landmark	<u>Account_ID</u>
--------------------	------------	-----------	------------	----------	----------	-------------------

3NF

Pin_Code	City	State
----------	------	-------

3NF

Database Implementation

```
mysql> SELECT * FROM adidas;
ERROR 1146 (42S02): Table 'adidas.adidas' doesn't exist
mysql> SHOW TABLES;
```

```
+-----+
| Tables_in_adidas |
+-----+
| account          |
| adidas_team      |
| belongs_to       |
| brand            |
| card             |
| category_has     |
| cod              |
| collection       |
| color            |
| country_of_origin |
| coupon           |
| customer         |
| customer_service |
| employee         |
| netbanking       |
| orders           |
| payment          |
| pincode          |
| product          |
| registered_id    |
| review           |
| section          |
| sports           |
| wallet           |
+-----+
25 rows in set (0.02 sec)
```

```
mysql> select * from adidas_team;
```

```
+-----+-----+
| Team_ID | Department_name |
+-----+-----+
|      1 | Import unit     |
|      2 | Customer Service unit |
|      3 | Verification unit |
|      4 | Website maintenance |
+-----+-----+
4 rows in set (0.20 sec)
```

```
mysql> select*from employee;
```

```
+-----+-----+-----+-----+-----+
| Employee_ID | Email_ID | Employee_name | Contact_no | Team_ID |
+-----+-----+-----+-----+-----+
| E001 | jim.halp22@adidas.com | Jim Halpert | 9456871322 | 1 |
| E002 | michael.scott1998@adidas.com | Michael Scott | 7895674992 | 4 |
| E003 | angela.martin@adidas.com | Angela Martin | 9124786322 | 2 |
| E004 | pamela.williams@adidas.com | Pamela Williams | 9845671232 | 3 |
| E005 | kelly.kapoor@adidas.com | Kelly Kapoor | 8794563215 | 1 |
| E006 | ryan.howard213@adidas.com | Ryan Howard | 9657412388 | 2 |
| E007 | ted.mosby@adidas.com | Ted Mosby | 7123654789 | 4 |
| E008 | amy.santiago@adidas.com | Amy Santiago | 9173216594 | 3 |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> select*from customer;
```

Customer_ID	First_name	Last_name	Contact_no	Pin_code	Landmark	Account_ID
A00001	Mansi	Kale	8794125638	411006	Lupin	A7360B0
A00002	Ritwik	Ahuja	8934820948	310323	City Center	A7360B6
A00003	Akanksha	Pore	9963217892	412587	TCS	A7360B3
A00004	Sahil	Sharma	7283909123	323862	Indira Stadium	A7360B8
A00005	Aman	Roy	9348945881	232455	City mall	A7360B5
A00006	Shivani	Singh	8897923485	394628	Gill garden	A7360B4
A00007	Pearl	Robert	6767329876	324897	Symbiosis	A7360B7
A00008	Simran	Kaur	9437936348	190290	Christ university	A7360B2
A00009	Sara	Khan	9823410100	238790	Link square	A7360B9
A00010	Udit	Khanna	7878394715	498764	Nehru Park	A7360B1

```
10 rows in set (1.72 sec)
```

```
mysql> select*from pincode;
```

Pin_code	City	State
190290	Bangalore	AP
232455	Indore	MP
238790	Mumbai	MH
310323	Jaipur	RJ
323862	Surat	GJ
324897	Noida	UP
394628	Dehradun	UK
411006	Pune	MH
412587	Delhi	Delhi
498764	Jodhpur	RJ

```
10 rows in set (0.00 sec)
```

```
mysql> select*from account;
```

Account_ID	Email_ID
A7360B0	mansi121@gmail.com
A7360B1	udit99k@gmail.com
A7360B2	kaursimran@gmail.com
A7360B3	apore@gmail.com
A7360B4	shivanisingh30@gmail.com
A7360B5	roy7801@gmail.com
A7360B6	ritzahuja@gmail.com
A7360B7	pvrobert@gmail.com
A7360B8	sahilsharma19@gmail.com
A7360B9	khansara009@gmail.com

```
10 rows in set (0.07 sec)
```

```
mysql> select*from registered_id;
```

Account_ID	Password
A7360B1	Pa@22984
A7360B2	88329#11122
A7360B4	SHIVANI9
A7360B5	R.o.Y@2323
A7360B6	33331212riwtik
A7360B8	Sahil@909@S

```
6 rows in set (0.40 sec)
```

```
mysql> select*from product;
```

P_ID	Name	Cost	Offer	material	B_ID	Team_ID	Collection_ID	Country_ID
PX001	Zeta 2.0 shoes	4599	30	Polyester	B011	1	110C1	C02
PX002	Copa shoes	5999	40	Synthetic leather	B011	1	110C3	C01
PX003	Retrurun	2199	NULL	Ripple rubber	B013	1	110C5	C03
PX004	Training Tee	2999	20	Elastane	B012	1	110C4	C02
PX005	Gym tracks	5299	40	Polyester	B012	1	110C8	C02
PX006	Striped Tights	4599	30	Recycled Polyester	B011	1	110C7	C05
PX007	Running jacket	6000	30	Polyamide	B013	1	110C9	C03
PX008	Camo backpack	1499	50	Polyester	B011	1	110C5	C04
PX009	Linear Duffle Bag	1499	40	Recycled Polyester	B013	1	110C6	C04
PX010	Ankle Socks	299	NULL	Cotton	B011	1	110C1	C02

```
10 rows in set (0.00 sec)
```

```
mysql> select*from color;
```

P_ID	color
PX001	White
PX001	Black
PX002	Blue
PX003	Dark Grey
PX004	Neon Green
PX004	White
PX005	Red
PX005	Navy Blue
PX006	Grey
PX007	Red
PX007	Neon Pink
PX007	Grey
PX008	Yellow
PX009	Camo
PX010	Black

```
15 rows in set (0.07 sec)
```

```
mysql> select*from collection;
```

Collection_ID	name
110C1	Ultraboost
110C2	Predator
110C3	Nemeziz
110C4	Copa
110C5	X
110C6	Home of classics
110C7	adicolor
110C8	Nite Jogger
110C9	NMD

```
9 rows in set (0.00 sec)
```

```
mysql> select*from belongs_to;
```

P_ID	Category_ID	Section_ID	Sport_ID
PX001	C2	S01	SP04
PX002	C2	S01	SP01
PX003	C2	S03	SP06
PX004	C1	S02	SP06
PX005	C1	S01	SP09
PX006	C1	S02	SP09
PX007	C1	S01	SP06
PX008	C3	S01	SP05
PX009	C3	S01	NULL
PX010	C3	S03	NULL

```
10 rows in set (1.70 sec)
```

```
mysql> select*from brand;
```

Brand_ID	name
B011	Adidas Originals
B021	Y-3
B031	Stella McCartney

```
3 rows in set (0.00 sec)
```

```
mysql> select*from card;
```

Payment_id	CWV	Name_on_card	Expiry	Card_no
U001	201	Mansi Kale	2012-06-24	1254 6513 3625 1025
U002	569	Ritwik Ahuja	2026-01-23	3569 5201 3327 4589

2 rows in set (1.71 sec)

```
mysql> select*from cart;
```

Cart_ID
KR00
KR01
KR02
KR03
KR04
KR05
KR06
KR07
KR08
KR09

10 rows in set (0.00 sec)

```
mysql> select*from category_has;
```

Category_ID	Section_ID	Sport_ID
C1	S01	SP01
C1	S02	SP01
C1	S03	SP05
C2	S01	SP04
C2	S02	SP03
C2	S04	SP05
C3	S01	SP06
C3	S02	SP08
C3	S02	SP06

9 rows in set (0.00 sec)

```
mysql> select*from cod;
+-----+-----+
| Payment_id | OTP    |
+-----+-----+
| U005       | 126587 |
+-----+-----+
1 row in set (0.18 sec)
```

```
mysql> select*from country_of_origin;
+-----+-----+
| Country_ID | name    |
+-----+-----+
| C01        | Philippines |
| C02        | India     |
| C03        | Vietnam  |
| C04        | Cambodia |
| C05        | Myanmar  |
+-----+-----+
5 rows in set (0.30 sec)
```

```
mysql> select*from customer_service;
+-----+-----+-----+-----+
| Cs_ID | Email_ID                               | Contact_no | Team_ID |
+-----+-----+-----+-----+
| CSER01 | customercare@adidas.com             | 888556635  | 2       |
| CSER02 | adidascustomers@adidas.com          | 969791100  | 2       |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select*from netbanking;
+-----+-----+
| Payment_id | Bank_name |
+-----+-----+
| U004       | Axis Bank |
+-----+-----+
1 row in set (0.07 sec)
```



```
mysql> select*from orders;
```

order_id	Discount	Subtotal	Status	Estimated_delivery	cart_ID	Coupon_id
TRX01	40	5999	Shipped	2005-12-21	KR02	1003
TRX02	20	2999	Received	2024-01-21	KR05	1002
TRX03	30	6499	Delivered	2008-02-21	KR03	1005
TRX04	50	9499	Dispatched	2019-01-21	KR01	1001
TRX05	35	4999	On the way	2021-02-21	KR04	1004

5 rows in set (0.00 sec)

```
mysql> select*from review;
```

Rev_ID	No_of_stars	Team_ID	Customer_ID
101	3	2	A00004
102	4	2	A00001
103	4	2	A00008
104	5	2	A00003
105	3.5	2	A00002

5 rows in set (0.03 sec)

```
mysql> select*from section;
```

Section_ID	name
S01	Men
S02	Women
S03	Girls
S04	Boys

4 rows in set (0.07 sec)

```
mysql> select*from sports;
```

```
+-----+-----+  
| Sport_ID | name      |  
+-----+-----+  
| SP01     | Basketball|  
| SP02     | Cricket   |  
| SP03     | Cycling   |  
| SP04     | Football  |  
| SP05     | Outdoor   |  
| SP06     | Running   |  
| SP07     | Swimming  |  
| SP08     | Tennis    |  
| SP09     | Training  |  
+-----+-----+  
9 rows in set (0.07 sec)
```

```
mysql> select*from wallet;
```

```
+-----+-----+  
| Payment_id | App_name |  
+-----+-----+  
| U003       | Paytm    |  
+-----+-----+  
1 row in set (0.07 sec)
```

Query Execution:

1)

```
mysql> select customer_id from review where no_of_stars=4;
+-----+
| customer_id |
+-----+
| A00001      |
| A00008      |
+-----+
2 rows in set (0.30 sec)
```

2)

```
mysql> select color.color, p_id from color group by color;
+-----+-----+
| color      | p_id |
+-----+-----+
| White      | PX001 |
| Black      | PX001 |
| Blue       | PX002 |
| Dark Grey  | PX003 |
| Neon Green | PX004 |
| Red        | PX005 |
| Navy Blue  | PX005 |
| Grey       | PX006 |
| Neon Pink  | PX007 |
| Yellow     | PX008 |
| Camo       | PX009 |
+-----+-----+
11 rows in set (0.00 sec)
```

3)

```
mysql> select * from color order by color;
```

P_ID	color
PX001	Black
PX010	Black
PX002	Blue
PX009	Camo
PX003	Dark Grey
PX006	Grey
PX007	Grey
PX005	Navy Blue
PX004	Neon Green
PX007	Neon Pink
PX005	Red
PX007	Red
PX001	White
PX004	White
PX008	Yellow

15 rows in set (0.17 sec)

4)

```
mysql> Select product.name, cost, material from product where cost=4599;
```

name	cost	material
Zeta 2.0 shoes	4599	Polyester
Striped Tights	4599	Recycled Polyester

2 rows in set (0.52 sec)

5)

```
mysql> select product.name from product where collection_id in(select collection_id from collection where name='x');
```

name
Retrorun
Camo backpack

2 rows in set (0.32 sec)

6)

```
mysql> select first_name,last_name from customer where customer_id in(select customer_id from review where no_of_stars=4);
+-----+-----+
| first_name | last_name |
+-----+-----+
| Mansi      | Kale      |
| Simran     | Kaur      |
+-----+-----+
2 rows in set (0.00 sec)
```

7)

```
mysql> select email_id from account where account_id in(select account_id from registered_id where password='Sahil@909@S');
+-----+
| email_id |
+-----+
| sahilsharma19@gmail.com |
+-----+
1 row in set (0.34 sec)
```

8)

```
mysql> select password from registered_id where account_id in(select account_id from account where email_id='shivanisingh30@gmail.com');
+-----+
| password |
+-----+
| SHIVANI9 |
+-----+
1 row in set (0.00 sec)
```

9)

```
mysql> select status from orders where cart_id in(select cart_id from cart where cart_id='KR02');
+-----+
| status |
+-----+
| Shipped |
+-----+
1 row in set (0.19 sec)
```

10)

```
mysql> select estimated_delivery from orders where extract(year from estimated_delivery)='2021';
+-----+
| estimated_delivery |
+-----+
| 2021-02-21         |
+-----+
1 row in set (0.07 sec)
```

11)

```
mysql> select card_no,name_on_card from card where cvv=201;
+-----+-----+
| card_no          | name_on_card |
+-----+-----+
| 1254 6513 3625 1025 | Mansi Kale   |
+-----+-----+
1 row in set (0.00 sec)
```

12)

```
mysql> select date_of_payment from payment where order_id in(select order_id from orders where order_id='TRX02');
+-----+
| date_of_payment |
+-----+
| 2005-12-20      |
+-----+
1 row in set (1.81 sec)
```

13)

```
mysql> select * from product;
+-----+-----+-----+-----+-----+-----+-----+-----+
| P_ID | Name          | Cost | Offer | material          | B_ID | Team_ID | Collection_ID | Country_ID |
+-----+-----+-----+-----+-----+-----+-----+-----+
| PX001 | Zeta 2.0 shoes | 4599 | 30    | Polyester         | B011 | 1       | 110C1         | C02        |
| PX002 | Copa shoes     | 5999 | 40    | Synthetic leather | B011 | 1       | 110C3         | C01        |
| PX003 | Retrorun       | 2199 | NULL  | Ripple rubber     | B013 | 1       | 110C5         | C03        |
| PX004 | Training Tee   | 2999 | 20    | Elastane          | B012 | 1       | 110C4         | C02        |
| PX005 | Gym tracks     | 5299 | 40    | Polyester         | B012 | 1       | 110C8         | C02        |
| PX006 | Striped Tights | 4599 | 30    | Recycled Polyester | B011 | 1       | 110C7         | C05        |
| PX007 | Running jacket | 6000 | 30    | Polyamide         | B013 | 1       | 110C9         | C03        |
| PX008 | Camo backpack  | 1499 | 50    | Polyester         | B011 | 1       | 110C5         | C04        |
| PX009 | Linear Duffle Bag | 1499 | 40    | Recycled Polyester | B013 | 1       | 110C6         | C04        |
| PX010 | Ankle Socks    | 299  | NULL  | Cotton            | B011 | 1       | 110C1         | C02        |
+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.14 sec)
```

14)

```
mysql> select Name from product where material='Polyester';
+-----+
| Name          |
+-----+
| Zeta 2.0 shoes |
| Gym tracks     |
| Camo backpack  |
+-----+
3 rows in set (0.03 sec)
```

15)

```
mysql> select Name from product where material='Cotton';
+-----+
| Name          |
+-----+
| Ankle Socks   |
+-----+
1 row in set (0.00 sec)
```

16)

```
mysql> select Name from product where material='Recycled Polyester';
+-----+
| Name          |
+-----+
| Striped Tights |
| Linear Duffle Bag |
+-----+
2 rows in set (0.00 sec)
```

17)

```
mysql> select avg(cost) from product;
+-----+
| avg(cost) |
+-----+
| 3499.1000 |
+-----+
1 row in set (0.00 sec)
```

18)

```
mysql> select name, cost from product where cost = (select min(cost) from product);
+-----+-----+
| name          | cost |
+-----+-----+
| Ankle Socks   | 299  |
+-----+-----+
1 row in set (0.00 sec)
```

19)

```
mysql> select name, cost from product where cost = (select max(cost) from product);
+-----+-----+
| name      | cost |
+-----+-----+
| Running jacket | 6000 |
+-----+-----+
1 row in set (0.00 sec)
```

20)

```
mysql> select Name, min(offer) from product;
+-----+-----+
| Name      | min(offer) |
+-----+-----+
| Zeta 2.0 shoes | 20 |
+-----+-----+
1 row in set (0.00 sec)
```

21)

```
mysql> select avg(cost) from product;
+-----+
| avg(cost) |
+-----+
| 3499.1000 |
+-----+
1 row in set (0.00 sec)
```

22)

```
mysql> select name, cost from product where offer = (select max(offer) from product);
+-----+-----+
| name      | cost |
+-----+-----+
| Camo backpack | 1499 |
+-----+-----+
1 row in set (0.00 sec)
```


23)

```
mysql> select name,offer from product where offer = (select max(offer) from product);
+-----+-----+
| name          | offer |
+-----+-----+
| Camo backpack |    50 |
+-----+-----+
1 row in set (0.00 sec)

mysql> select name,offer from product where offer = (select min(offer) from product);
+-----+-----+
| name          | offer |
+-----+-----+
| Training Tee  |    20 |
+-----+-----+
1 row in set (0.00 sec)
```

24)

```
mysql> select Name from product where P_ID in (select P_ID from belongs_to where category_id in (select category_id from category where name='Clothing'));
+-----+
| Name          |
+-----+
| Training Tee  |
| Gym tracks    |
| Striped Tights |
| Running jacket |
+-----+
4 rows in set (0.06 sec)
```

25)

```
mysql> select Name from product where Country_ID='C03';
+-----+
| Name          |
+-----+
| Retrorun      |
| Running jacket |
+-----+
2 rows in set (0.00 sec)
```

26)

```
mysql> select * from color;
+-----+-----+
| P_ID | color |
+-----+-----+
| PX001 | White |
| PX001 | Black |
| PX002 | Blue  |
| PX003 | Dark Grey |
| PX004 | Neon Green |
| PX004 | White |
| PX005 | Red   |
| PX005 | Navy Blue |
| PX006 | Grey  |
| PX007 | Red   |
| PX007 | Neon Pink |
| PX007 | Grey  |
| PX008 | Yellow |
| PX009 | Camo   |
| PX010 | Black |
+-----+-----+
15 rows in set (0.01 sec)
```

27)

```
mysql> select adidas_team.team_id, adidas_team.department_name, employee.employee_id, employee.employee_name, employee.email_id
-> from adidas_team
-> inner join employee
-> on adidas_team.team_id=employee.team_id;
+-----+-----+-----+-----+-----+
| team_id | department_name | employee_id | employee_name | email_id |
+-----+-----+-----+-----+-----+
| 1 | Import unit | E001 | Jim Halpert | jim.halp22@adidas.com |
| 4 | Website maintenance | E002 | Michael Scott | michael.scott1998@adidas.com |
| 2 | Customer Service unit | E003 | Angela Martin | angela.martin@adidas.com |
| 3 | Verification unit | E004 | Pamela Williams | pamela.williams@adidas.com |
| 1 | Import unit | E005 | Kelly Kapoor | kelly.kapoor@adidas.com |
| 2 | Customer Service unit | E006 | Ryan Howard | ryan.howard213@adidas.com |
| 4 | Website maintenance | E007 | Ted Mosby | ted.mosby@adidas.com |
| 3 | Verification unit | E008 | Amy Santiago | amy.santiago@adidas.com |
+-----+-----+-----+-----+-----+
8 rows in set (0.17 sec)
```

Functions:

1) To get average review:

```
mysql> use adidas
Database changed
mysql> delimiter @@
mysql> create function avg_review()
    -> returns float
    -> begin
    -> declare avg float;
    -> select avg(no_of_stars) into avg from review;
    -> return avg;
    -> end @@
Query OK, 0 rows affected (0.37 sec)
```

```
mysql> select avg_review();
    -> \g
+-----+
| avg_review() |
+-----+
|          3.9 |
+-----+
1 row in set (0.22 sec)
```

2) To find maximum discount:

```
mysql> delimiter @@
mysql> create function max_discount()
    -> returns float
    -> begin
    -> declare max float;
    -> select max(discount) into max from orders;
    -> return max;
    -> end @@
Query OK, 0 rows affected (0.30 sec)
```

```
mysql> select max_discount();
-> \g
+-----+
| max_discount() |
+-----+
|          50 |
+-----+
1 row in set (0.19 sec)
```

3) To find minimum price:

```
mysql> delimiter @@
mysql> create function min_price()
-> returns float
-> begin
-> declare min float;
-> select min(cost) into min from product;
-> return min;
-> end @@
```

```
mysql> select min_price();
+-----+
| min_price() |
+-----+
| 299 |
+-----+
1 row in set (0.36 sec)
```

4) Finding first name of customer with last name="Roy"

```
mysql> delimiter $$
mysql> create function customer_name()
-> returns varchar(12)
-> deterministic
-> begin
-> declare name2 varchar(12);
-> select first_name into name2 from customer where last_name="Roy";
-> return (name2);
-> end $$
Query OK, 0 rows affected (0.64 sec)

mysql> select customer_name();
-> \g
+-----+
| customer_name() |
+-----+
| Aman |
+-----+
1 row in set (0.02 sec)
```

5) Delivery date of order with status shipped.

```
mysql> delimiter $$
mysql> create function delivery_date()
  -> returns date
  -> deterministic
  -> begin
  -> declare date1 date;
  -> select estimated_delivery into date1 from orders where Status="shipped";
  -> return (date1);
  -> end$$
Query OK, 0 rows affected (0.03 sec)
```



```
mysql> select delivery_date();
  -> \g
+-----+
| delivery_date() |
+-----+
| 2020-12-31      |
+-----+
1 row in set (0.00 sec)
```

Procedures

1.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `customer_name`()  
BEGIN  
  select first_name,last_name from customer;  
END
```


```
1 • call adidas.customer_name();  
2
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export: 		
Wrap Cell Content: 		
	first_name	last_name
▶	Mansi	Kale
	Ritwik	Ahuja
	Akanksha	Pore
	Sahil	Sharma
	Aman	Roy
	Shivani	Singh
	Pearl	Robert
	Simran	Kaur
	Sara	Khan
	Udit	Khanna

2.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `payment_mode`()  
BEGIN  
select status from orders where order_id in(select order_id from payment where payment_id  
in(select payment_id from COD) or payment_id=(select payment_id from netbanking));  
END
```

```
1 • call adidas.order_year();  
2
```

<  Filter Rows: Export

	order_id
▶	TRX05

3.

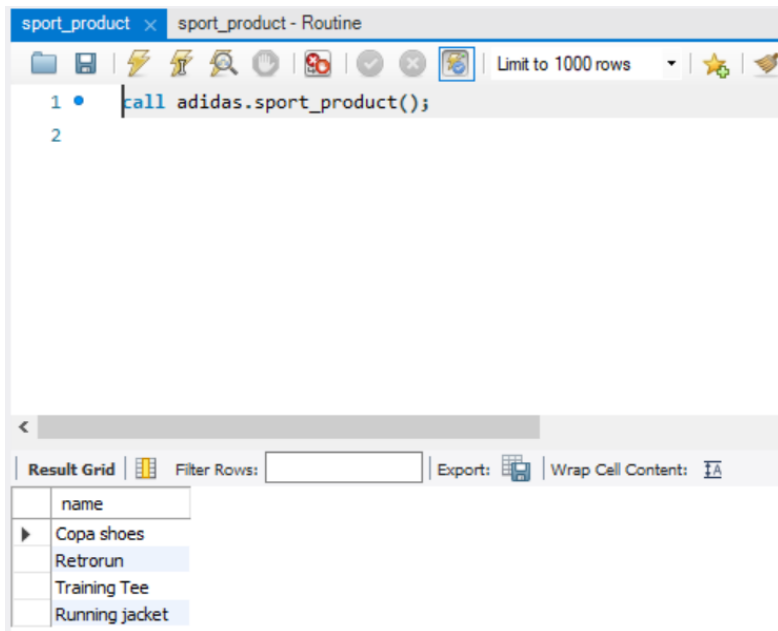
```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `colors`()  
2 BEGIN  
3 select * from product where P_ID in (select P_ID from color group by p_ID having count(p_ID)>1);  
4 END
```

```
1 • call adidas.colors();
2
```

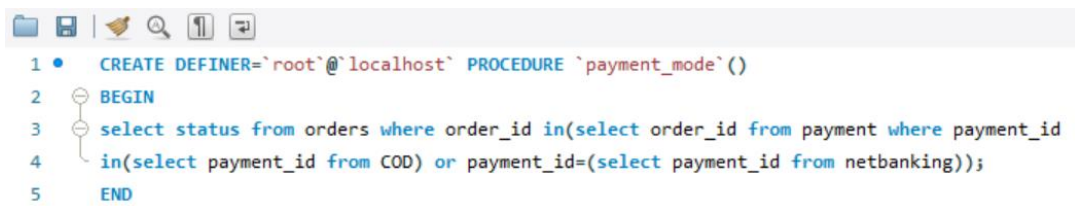
<									
Result Grid									
Filter Rows:									
Export:									
Wrap Cell Content:									
	P_ID	name	Cost	Offer	material	B_ID	Team_ID	Collection_ID	Country_ID
▶	PX001	Zeta 2.0 shoes	4599	30	Polyester	B011	1	110C1	C02
	PX004	Training Tee	2999	20	Elastane	B012	1	110C4	C02
	PX005	Gym tracks	5299	40	Polyester	B012	1	110C8	C02
	PX007	Running jacket	6000	30	Polyamide	B013	1	110C9	C03

4.

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `sport_product`()
2 BEGIN
3     select name from product p where p_id in
4     (select p_id from belongs_to where sport_id in
5     (select sport_id from sports where name="Running" or name="Basketball"));
6 END
```

5.



```
1 • call adidas.payment_mode();  
2
```

result Grid		Filter Rows:	Export:	Wrap Cel
status				
Shipped				
Delivered				

Triggers

1.

```
mysql> CREATE TRIGGER offer_add
-> before INSERT
-> on coupon
-> for each row
-> set new.amount=new.amount + 5;
Query OK, 0 rows affected (0.47 sec)
```

```
mysql> insert into coupon( Coupon_id,coupon_code,Amount,validity,Account_ID) values(1006,'Pg0672Y',10,'2020-08-11','A7360B9');
Query OK, 1 row affected (0.06 sec)
```

```
mysql> select * from coupon;
```

Coupon_id	coupon_code	Amount	validity	Account_ID
1001	Ax5286W	50	2024-12-20	A7360B0
1002	Mp6397N	20	2013-01-21	A7360B9
1003	Mh7946T	40	2010-12-20	A7360B2
1004	Fg7800R	35	2020-02-21	A7360B6
1005	Zc3628Y	30	2004-02-21	A7360B3
1006	Pg0672Y	15	2020-08-11	A7360B9

```
6 rows in set (0.02 sec)
```

2.

```
mysql> create trigger order_deduct
-> before update
-> on orders
-> for each row
-> set new.subtotal = new.subtotal + 1000;
Query OK, 0 rows affected (0.46 sec)
```

```
mysql> update orders set Estimated_Delivery = '2020/12/19';
Query OK, 5 rows affected (0.17 sec)
Rows matched: 5  Changed: 5  Warnings: 0
```

```
mysql> select * from orders;
```

order_id	Discount	Subtotal	Status	Estimated_delivery	cart_ID	Coupon_id
TRX01	40	6999	Shipped	2020-12-19	KR02	1003
TRX02	20	3999	Received	2020-12-19	KR05	1002
TRX03	30	7499	Delivered	2020-12-19	KR03	1005
TRX04	50	10499	Dispatched	2020-12-19	KR01	1001
TRX05	35	5999	On the way	2020-12-19	KR04	1004

5 rows in set (0.09 sec)

3.

```
mysql> create trigger selection
-> BEFORE Update ON cart_has_product
-> for each row
-> set new.no_of_items = new.no_of_items*2;
Query OK, 0 rows affected (1.62 sec)
```

```
mysql> update cart_has_product set no_of_items = 3 where no_of_items=1;
Query OK, 7 rows affected (0.35 sec)
Rows matched: 7  Changed: 7  Warnings: 0
```

```
mysql> select * from cart_has_product;
```

L_ID	Cart_ID	P_ID	Wishlist_ID	Date_placed	No_of_items	Size
1	KR01	PX003	W011	2020-11-12	6	5
2	KR01	PX005	W011	2020-11-13	2	S
3	KR01	PX006	NULL	NULL	6	M
4	KR02	PX001	W012	2020-11-23	6	6
5	KR02	PX010	W012	2020-11-24	3	M
6	KR03	PX004	NULL	NULL	6	S
7	KR04	PX009	W014	2020-11-26	2	NS
8	KR05	PX007	NULL	NULL	6	L
9	NULL	PX002	W016	2020-05-12	6	7
10	NULL	PX005	W019	2020-12-14	6	L

10 rows in set (0.11 sec)

4.

```
mysql> create trigger offer_update
-> before update
-> on product
-> for each row
-> set new.offer = old.offer - new.offer;
Query OK, 0 rows affected (0.12 sec)

mysql> update product set offer=30 where name="Camo backpack";
Query OK, 1 row affected (0.04 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from product;
```

P_ID	name	Cost	Offer	material	B_ID	Team_ID	Collection_ID	Country_ID
PX001	Zeta 2.0 shoes	4599	30	Polyester	B011	1	110C1	C02
PX002	Copa shoes	5999	40	Synthetic leather	B011	1	110C3	C01
PX003	Retrurun	2199	NULL	Ripple rubber	B013	1	110C5	C03
PX004	Training Tee	2999	20	Elastane	B012	1	110C4	C02
PX005	Gym tracks	5299	40	Polyester	B012	1	110C8	C02
PX006	Striped Tights	4599	30	Recycled Polyester	B011	1	110C7	C05
PX007	Running jacket	6000	30	Polyamide	B013	1	110C9	C03
PX008	Camo backpack	1499	20	Polyester	B011	1	110C5	C04
PX009	Linear Duffle Bag	1499	40	Recycled Polyester	B013	1	110C6	C04
PX010	Ankle Socks	299	NULL	Cotton	B011	1	110C1	C02

```
10 rows in set (0.00 sec)
```

5.

```
mysql> create trigger offer_change
-> before update
-> on product
-> for each row set new.offer = new.offer + 10;
Query OK, 0 rows affected (0.19 sec)

mysql> update product set offer=30 where name="Ankle Socks";
Query OK, 0 rows affected (0.16 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> select * from product;
```

P_ID	name	Cost	Offer	material	B_ID	Team_ID	Collection_ID	Country_ID
PX001	Zeta 2.0 shoes	4599	30	Polyester	B011	1	110C1	C02
PX002	Copa shoes	5999	40	Synthetic leather	B011	1	110C3	C01
PX003	Retrurun	2199	NULL	Ripple rubber	B013	1	110C5	C03
PX004	Training Tee	2999	20	Elastane	B012	1	110C4	C02
PX005	Gym tracks	5299	40	Polyester	B012	1	110C8	C02
PX006	Striped Tights	4599	30	Recycled Polyester	B011	1	110C7	C05
PX007	Running jacket	6000	30	Polyamide	B013	1	110C9	C03
PX008	Camo backpack	1499	20	Polyester	B011	1	110C5	C04
PX009	Linear Duffle Bag	1499	40	Recycled Polyester	B013	1	110C6	C04
PX010	Ankle Socks	299	NULL	Cotton	B011	1	110C1	C02

```
10 rows in set (0.00 sec)
```