

# Database for Web Developers

Ideas for today and tomorrow

Dr. Ayman Ezzat

Assistant Professor – MIU, FCIH

# A database in real practice

+ Options

UserName	FullIName	Password	Hobby	Address	Tel1	Tel2	Orders
Ahmed	Ahmed Mohamed Ibrahim	123456	Tennis, FootBall	Egypt	010		Mobile, X12, 350 EGP, 22/12/2017. Mobile, Z4, 240...
Ahmed	Ahmed Mohamed Hussein	123	Tenis, Karate	Egypt	010	012	N/A
Kareem	Kareem Hani Hossam	4568	Tennis	Egypt, Cairo, Nasr City, Makram Ebid	010123		Pencil, 300 EGP, 22/1/2018. Mobile, 250 Egp, 22/1...
Laila	Laila Hussein	9898	Tennis, BasketBall, reading	madint Nasr, Cairo, Misr			Mobile X121, 240 Egp, 22/2/2017

- Answer the following Questions

- How many items bought by username Ahmed?
  - Ahmed username is repeated twice, hmm which one ?
- How many users play Tennis ?
  - Is it Tennis or Tenis ? Should I used select \* from where = !!!
- How many users Living in Nasr City?
  - Someone write his address in franko arab !!!
- How many users Bought x12 Mobile?
  - Is it x12 or x121 how you can distinguish like %%
- Can you edit the row number 3 ?
  - There is not ID for each row how can you fetch this row
- What are the telephone numbers for user Kareem?
  - How about Null
- What is the password for user Laila?
  - Wow all passwords are available for everyone !!!!

← Server: 127.0.0.1 » Database: E-Shop » Table: Users

Browse Structure SQL Search



Table structure Relation view


#	Name	Type	Collation	Attributes	Null
<input type="checkbox"/>	1 <b>UserName</b>	varchar(255)	latin1_swedish_ci		No
<input type="checkbox"/>	2 <b>FullIName</b>	varchar(255)	latin1_swedish_ci		No
<input type="checkbox"/>	3 <b>Password</b>	varchar(255)	latin1_swedish_ci		No
<input type="checkbox"/>	4 <b>Hobby</b>	varchar(255)	latin1_swedish_ci		No
<input type="checkbox"/>	5 <b>Address</b>	varchar(255)	latin1_swedish_ci		No
<input type="checkbox"/>	6 <b>Tel1</b>	varchar(255)	latin1_swedish_ci		No
<input type="checkbox"/>	7 <b>Tel2</b>	varchar(255)	latin1_swedish_ci		No
<input type="checkbox"/>	8 <b>Orders</b>	text	latin1_swedish_ci		No

# 12 Normalizations – beyond theoretical issues

1. Never have a table without an ID.
2. Never Copy and Paste Fields !.
3. Strings must not be repeated.
4. NO Nulls in the Database by any means.
5. A comma separated Field is the worst to do in your life.
6. 4 tables are always lovely and will save you many times.
7. Working with Int (Numbers) is always accurate and preferred.
8. Use EAV whenever attributes are changing
9. Use the Self reference key for any tree.
10. We hate the like statement as it is always very slow
11. Always Fill your Tables with Data so you understand your mistakes
12. Never delete a record physically from Database always have log for all transactions.

# Creating Database issues

 **Create database** 

VideoDB Support Multi Language  
utf8\_general\_ci 

**Create**

# 1 Duplicated Values – Split by ID

```
SELECT * FROM `users` WHERE `UserName` = 'Ahmed'
```

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

UserName	FullName	Password	Hobby	Address	Tel1	Tel2	Orders
Ahmed	Ahmed Mohamed Ibrahim	123456	Tennis, FootBall	Egypt	010	012	Mobile, X12, 350 EGP, 22/12/2017. Mobile, Z4, 240...
Ahmed	Ahmed Mohamed Hussein	123	Tenis, Karate	Egypt	010	012	N/A

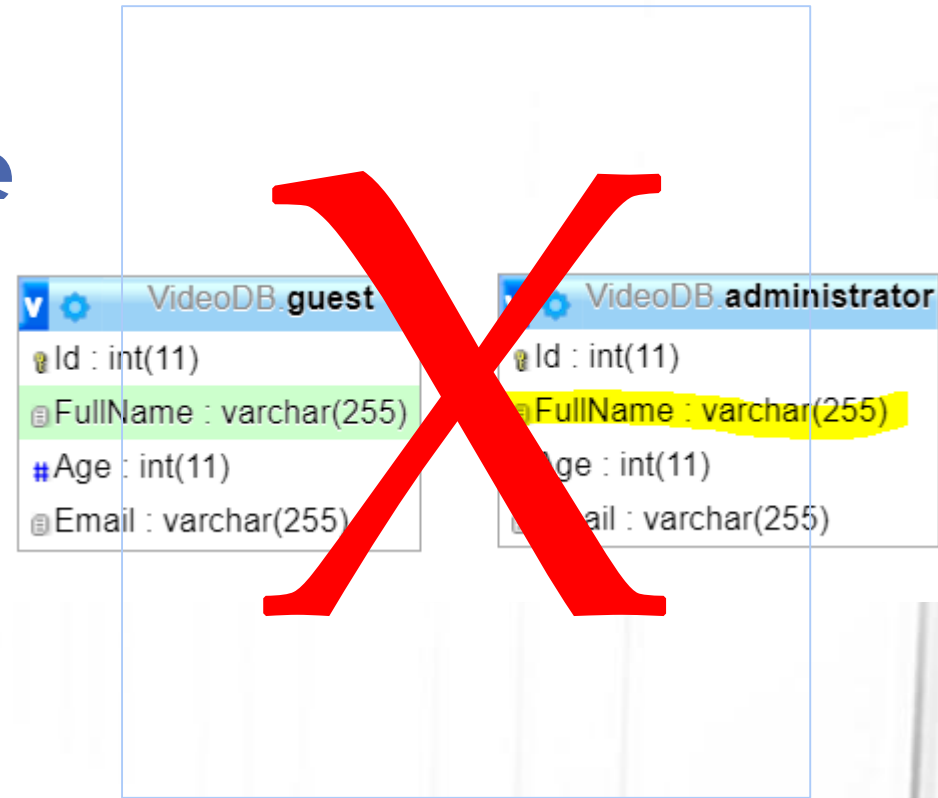
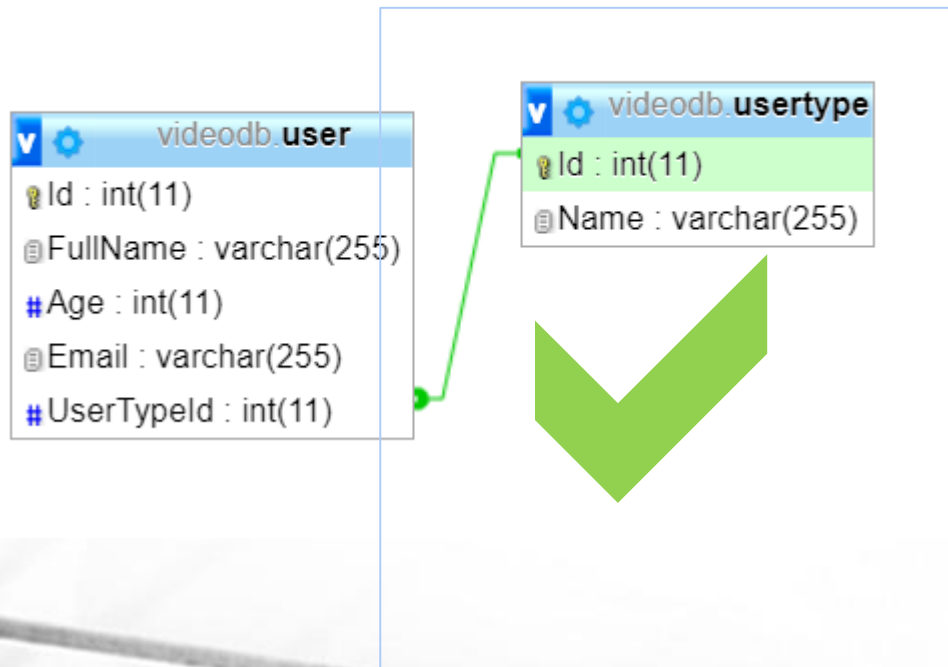
+ Options

	Id	FullName	UserType_ID
<input type="checkbox"/> Edit Copy Delete	1	Ahmed	1
<input type="checkbox"/> Edit Copy Delete	2	Kareem	2

Always use ID for any table, you will use this ID to catch the ROW

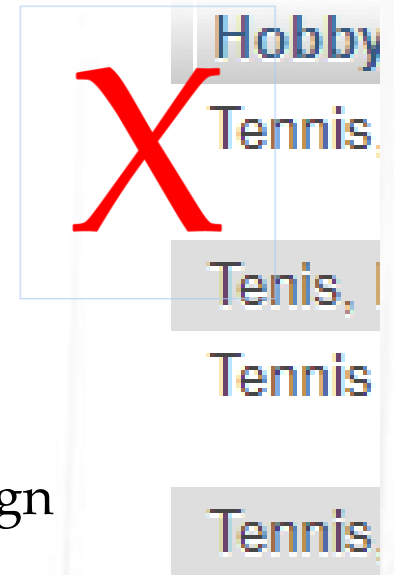
## 2 Never Copy and paste

- Duplicate results must be collected in one table and then distinguished by ID as foreign key with restrictions if needed.



## 3 Strings must not be repeated

- Repeated Strings Waste Space
- Results of any search is not accurate and will have mistakes
- Words could have spelling mistakes
- Solution, Take any repeated string into a sperate table and have foreign key to this table so simply repeat numbers.



# 4 NO Nulls in the Database by any means.

- Null make querying database with huge amount of data very slow
- Null Waste Space
- Null makes your code very hard to handle when you have empty strings in the same time
- **To solve it:**
- Take whatever it has null outside in a new table

student tel	
Id	: int(11)
Tel	: varchar(255)
UserId	: int(11)

student.tb_users	
ID	: int(11)
FullName	: varchar(255)
Email	: varchar(255)
DOB	: varchar(255)
Password	: varchar(255)
UserType_id	: int(11)

# 5 A comma separated Field is the worst to do in your life Many Many

- Any separator in input data means you have many to many relations' most probably hence
- Split into 2 tables and put connecting table

student.student	
Id : int(11)	
FullName : varchar(255)	
Address : varchar(255)	

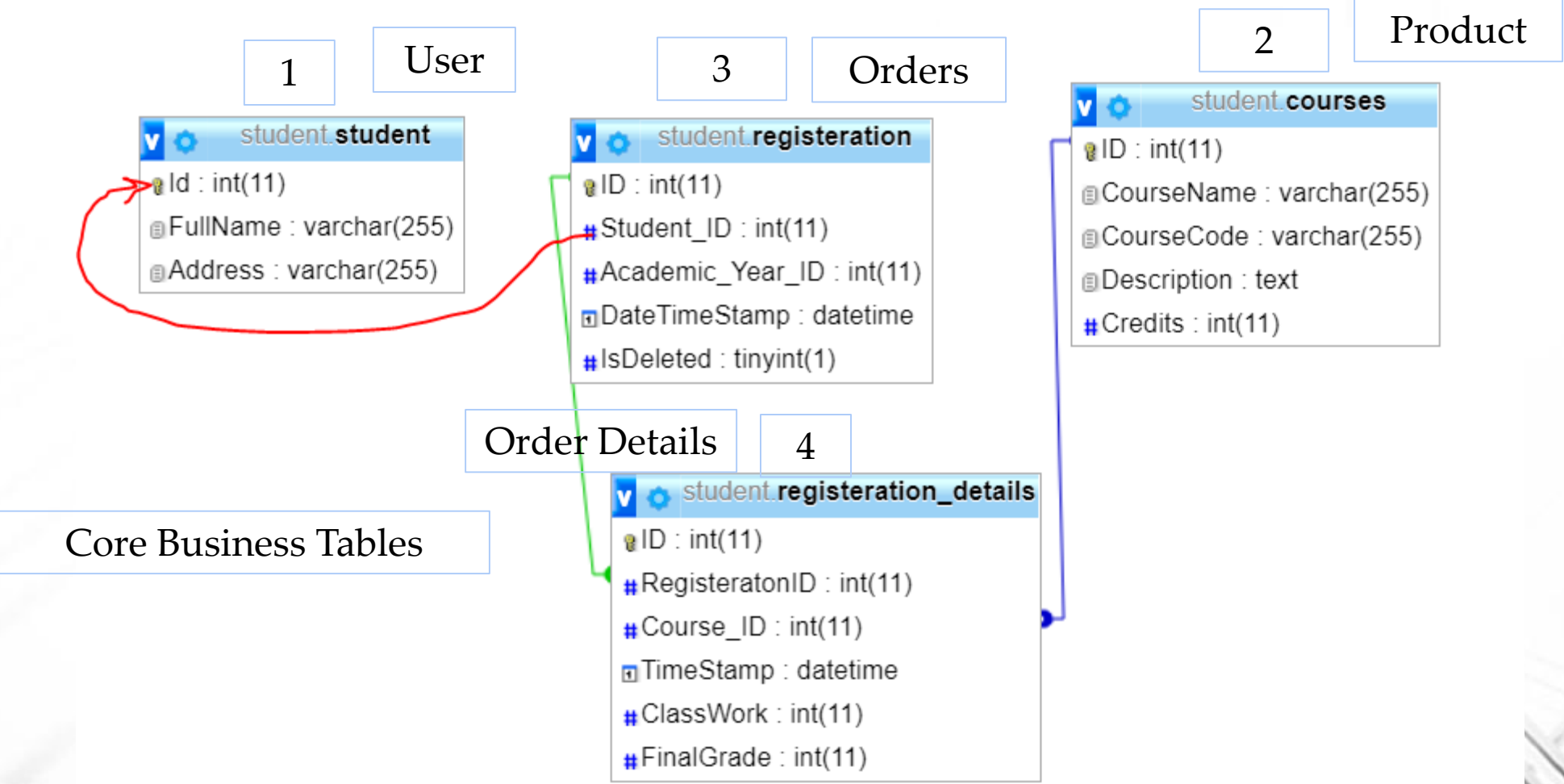
student.hobbies	
Id : int(11)	
HobbyName : varchar(255)	

student.studenthobby	
Id : int(11)	
StudentID : int(11)	
HobbyID : int(11)	

Hobby	
Tennis, FootBall	
Tennis, Karate	
Tennis	
Tennis, BasketBall, reading	

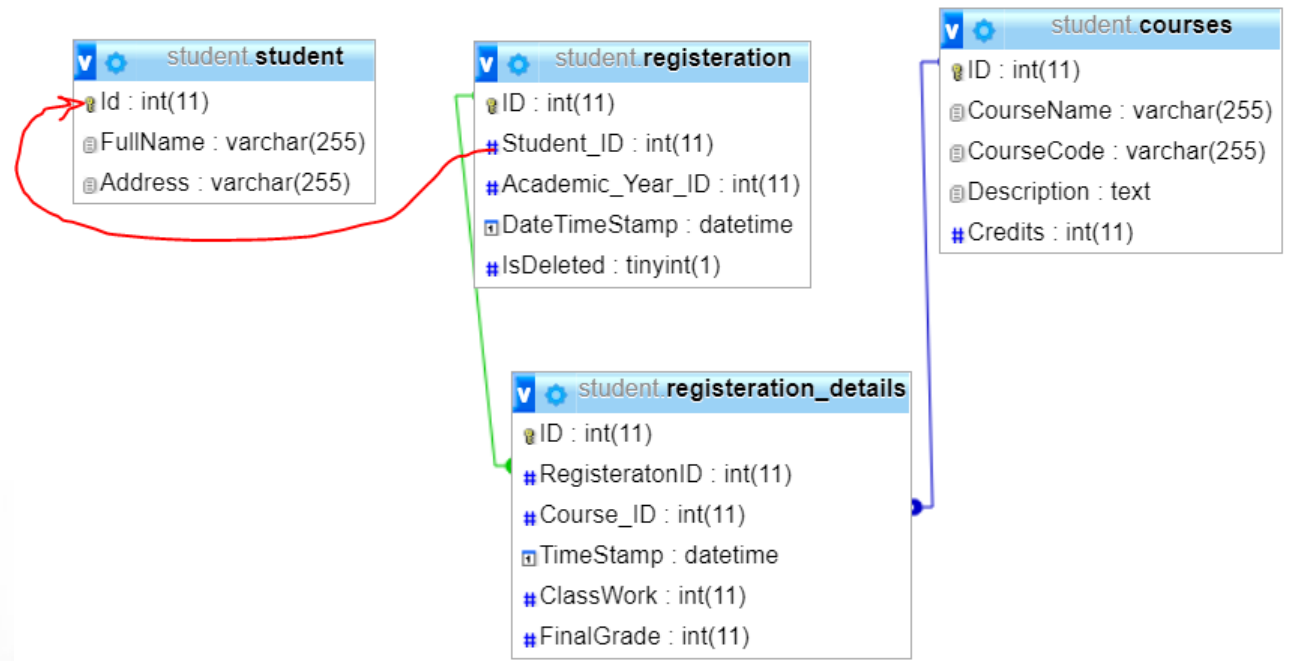
Many to Many Relationship is always preferred and gives your site flexibility if requirements changed.

# 6- 4 tables are always lovely and will save you many times – Core Business



# 7 Working with Int (Numbers) is always accurate and preferred

- Select \* from anytable where Id=.....
- Use all time the equal as it is always accurate and never have mistakes
- Tell me who takes the course “Decision making”
- Select \* from reg\_details where Course\_Id=1



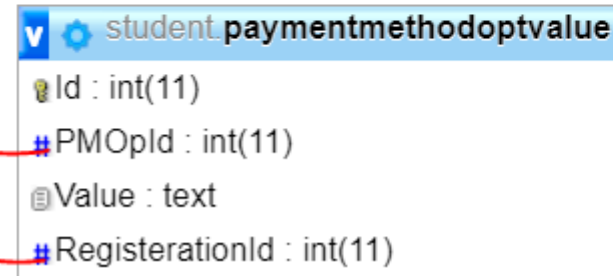
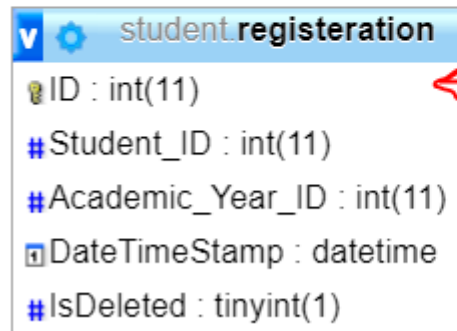
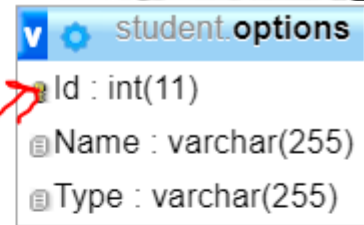
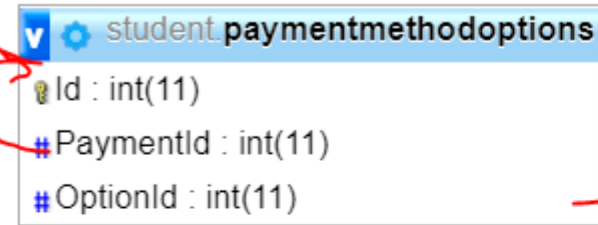
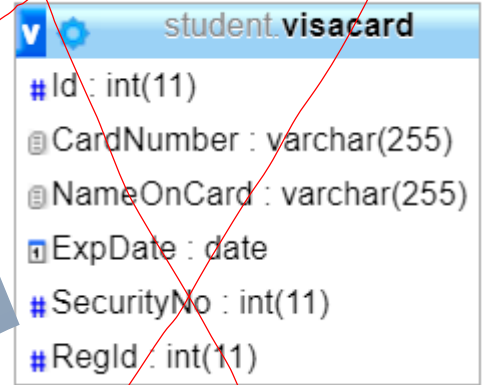
# 8 Use EAV whenever attributes are changing

- Payment Method are changing frequently
- Never have table with Null Columns
- Never have a table per each payment method !!!! You ill end up with infinite tables for each payment method what happens if new payment method appears !

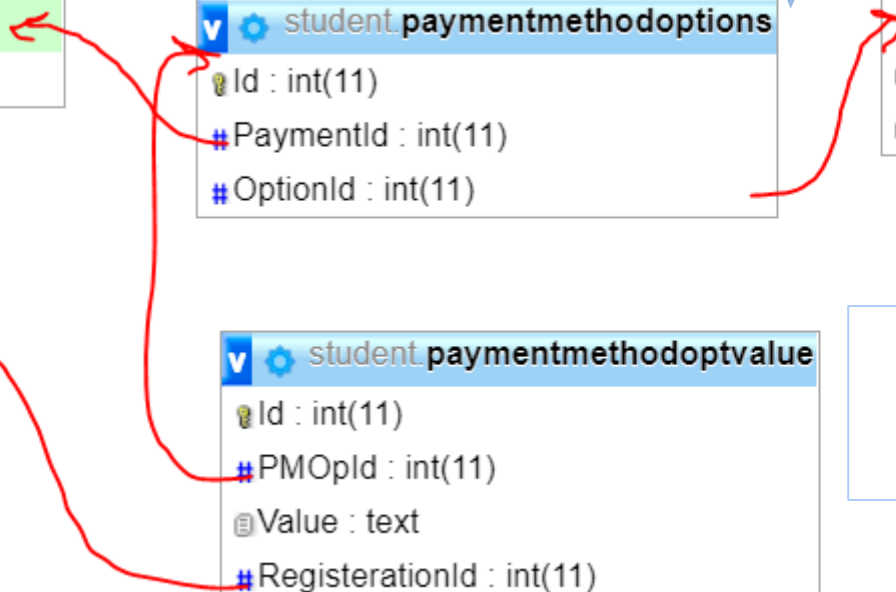
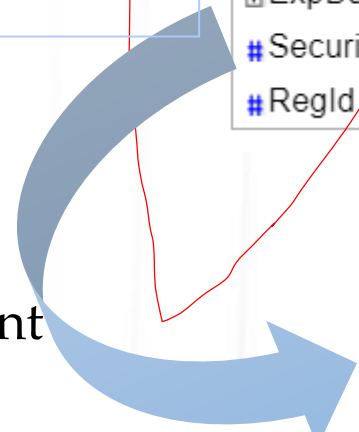
## Apply to

- Product Types
- User types
- Course Grade Types
- Forms !!!!

Convert Col to rows

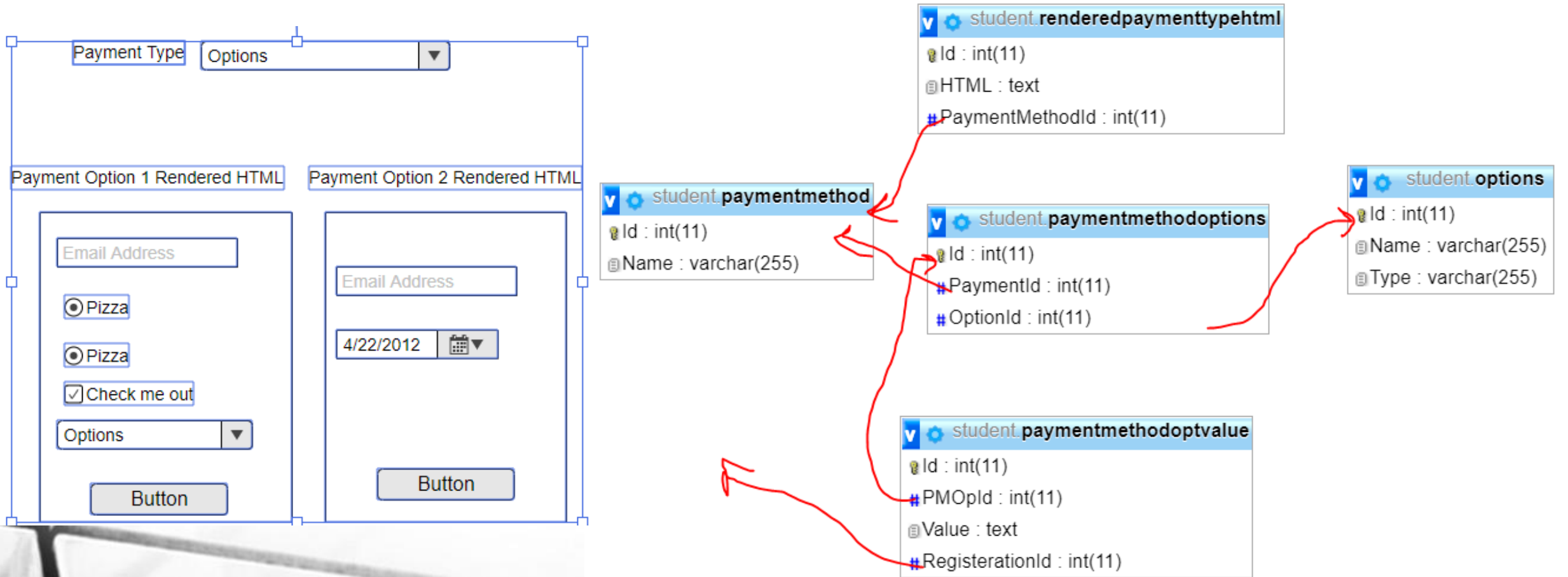


Flexibility Versus Code Complexity



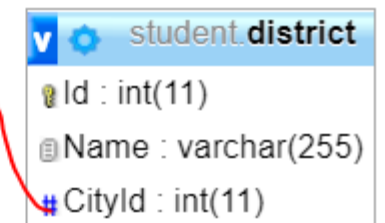
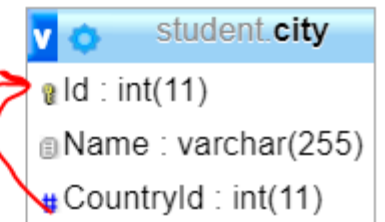
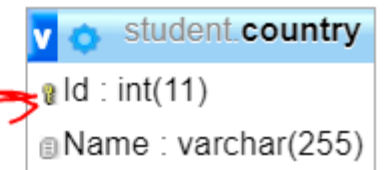
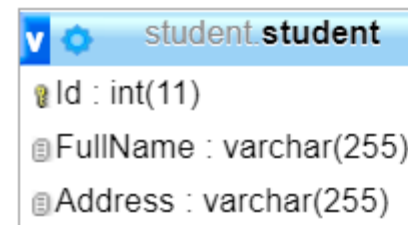
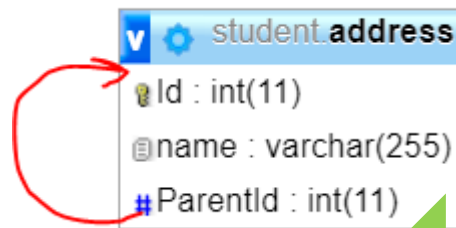
# Enhanced EAV

- Keep copy of rendered HTML on a separate table instead of many nested join statements.
- Update the Copy only when you update your system with new options



# 9 Use the Self reference key for any tree

- Putting address as varchar will leads to spelling mistakes
- Very difficult to get accurate results.
- If we make a multi foreign key (nested) then **change** requirements you will end by adding new table so you change **structure** and your **code** will change too. Hence you end with bad design.
- Use self reference key with any tree in your DB
  - Links
  - User Types
  - Courses
  - Product Types



A screenshot of a web application showing a list of addresses. The first two entries are "Egypt" and "Egypt, Cairo, Nasr City, Makram Ebid", and the third is "madint Nasr, Cairo, Misr". A large red 'X' is drawn over the first two entries, indicating that using a single varchar field for addresses leads to inconsistent and messy data.

# 10 We hate the like statement as it is always very slow

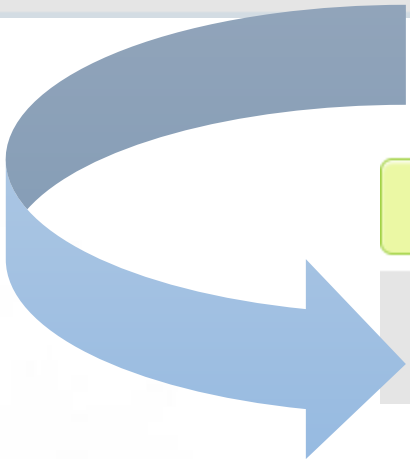
Show query box

✔ Showing rows 0 - 24 (248 total, Query took 0.0951 seconds.)

```
SELECT * FROM `employees` WHERE `first_name` like '%Gino%'
```

✔ Showing rows 0 - 24 (248 total, Query took 0.0829 seconds.)

```
SELECT * FROM `employees` WHERE `first_name` = 'Gino'
```

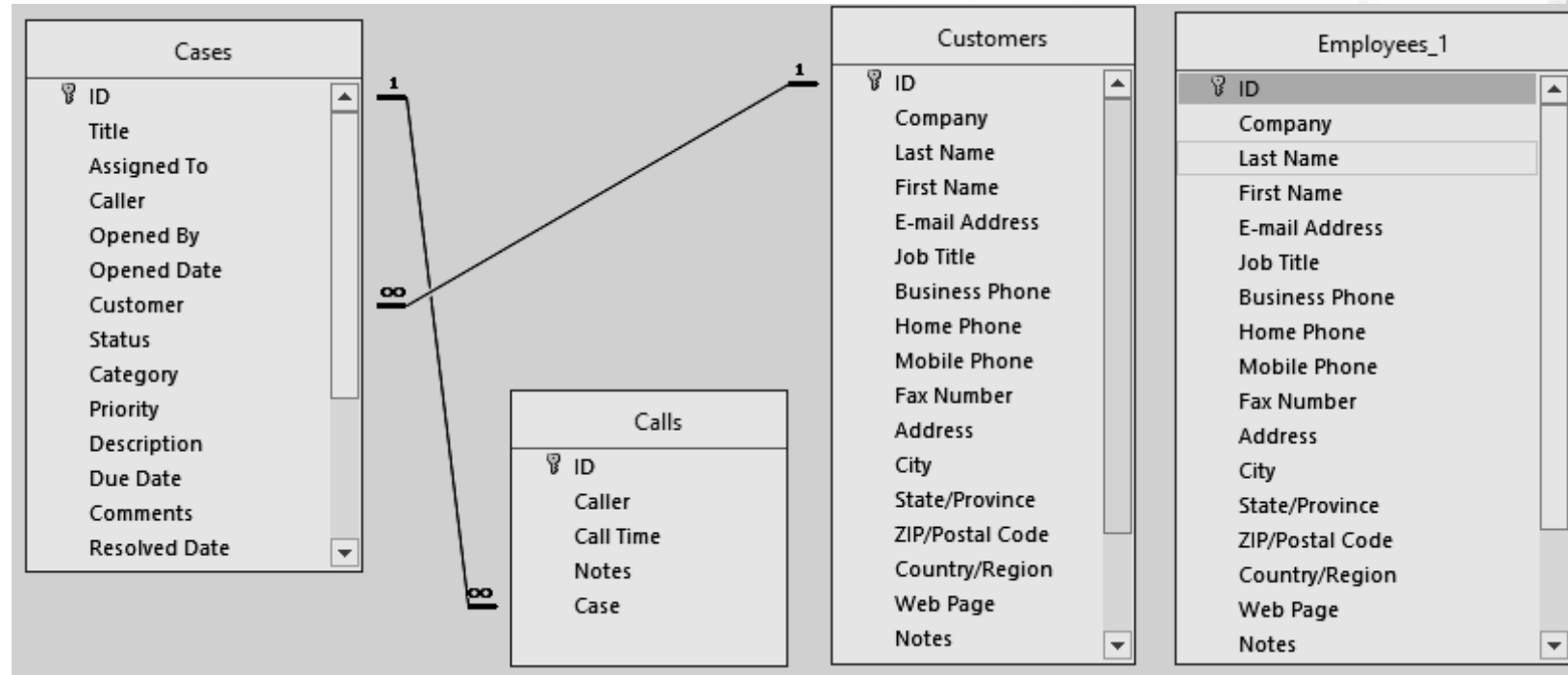


✔ Showing rows 0 - 0 (1 total, Query took 0.0379 seconds.)

```
SELECT * FROM `employees` WHERE `emp_no` = 10063
```

# 11 Always Fill your Tables with Data so you understand your mistakes

- Cases
- Assigned to (Many Many)
- Caller (Lookup Table)
- Customer (Many Many)
- Status (Lookup Table)
- Category (Lookup Table)
- Priority (Lookup Table)
- Comments (Self Ref)



# 12 Always have log for all transactions

- Keep tracking of who did what on your rows
- Each table must have this fields
  - Isdeleted (for Reports)
  - Createdat
  - Updated at
- Each database must keep log of all transactions in one table with status and datetime stamp.

student_users	
id	int(10) unsigned
name	varchar(255)
email	varchar(255)
password	varchar(255)
remember_token	varchar(100)
created_at	timestamp
updated_at	timestamp

student_logtransactions	
Id	int(11)
TransactionMessage	text
StatusId	int(11)
DateTimeStamp	timestamp

student_tb_users	
ID	int(11)
FullName	varchar(255)
Email	varchar(255)
DOB	varchar(255)
Password	varchar(255)
UserType_id	int(11)
IsDeleted	tinyint(1)
LastUpdated	timestamp

student_transactionslog	
Id	int(11)
Message	text
StatusId	int(11)
UserId	int(11)

# Patterns in Database 1 – Multi Role

- Any web system has multirole and permissions feature.
- Put HTML inside Pages so you can search anytime
  - Search
  - Update



# Patterns in Database 2- Multi-Lingual

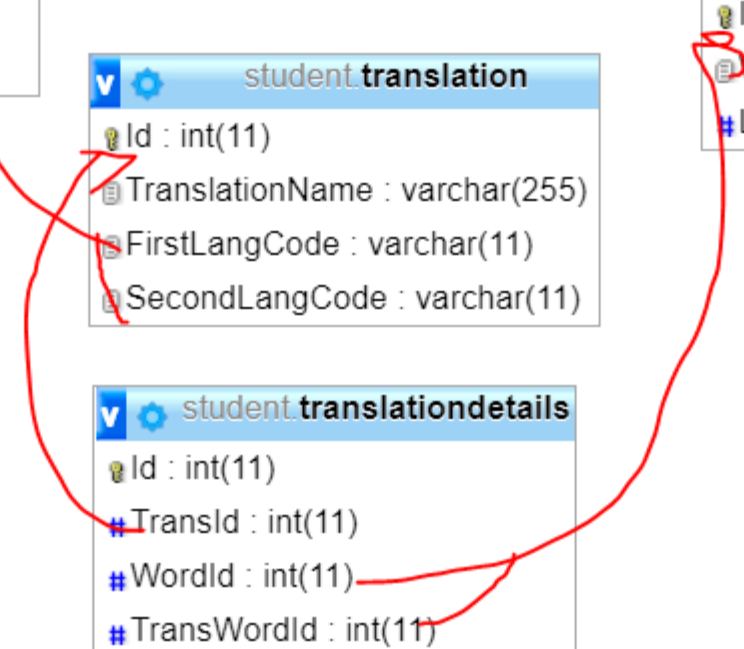
- Again 4 tables
- Add any language any time
- Add as much words as possible
- Difficult to implement but assure flexibility

student.languages	
Id : int(11)	
Name : int(11)	

student.translation	
Id : int(11)	
TranslationName : varchar(255)	
FirstLangCode : varchar(11)	
SecondLangCode : varchar(11)	

student.word	
Id : int(11)	
Word : varchar(255)	
LanguageId : int(11)	

student.translationdetails	
Id : int(11)	
TransId : int(11)	
WordId : int(11)	
TransWordId : int(11)	



# Patterns in Database 3 - Reporting

- Customized Reports easy to store SQL statement in side the Database

Search by Province | Search by Station Name | **Search by Proximity**

[How to Use - Search by Station Name](#)

Name:

contains     begins with

for years from:    1840 ▼ to 2013 ▼

or a specific date:    2013 ▼    August ▼    13 ▼

Display  results per page.

**student.tb\_users**

- ID : int(11)
- FullName : varchar(255)
- Email : varchar(255)
- DOB : varchar(255)
- Password : varchar(255)
- #UserType\_id : int(11)
- #IsDeleted : tinyint(1)
- LastUpdated : timestamp

**student.customizedreport**

- Id : int(11)
- ReportName : varchar(255)
- #UserId : int(11)
- SQLStatement : text**

# Patterns in DB -4 Hashing

- Locking Table Columns ,rows from changes
- Passwords
- Pass ID from page to another instead of ID

RegisteratonID	Course_ID	TimeStamp	ClassWork	FinalGrade
6	3	2016-04-26 11:25:12	15	25
6	4	2016-04-26 11:25:12	15	25
7	4	2016-04-26 14:17:36	15	25
7	6	2016-04-26 14:17:36	15	25
8	1	2016-04-28 11:27:47	15	25
8	3	2016-04-28 11:27:47	15	25
8	2	2016-04-28 11:27:48	15	25

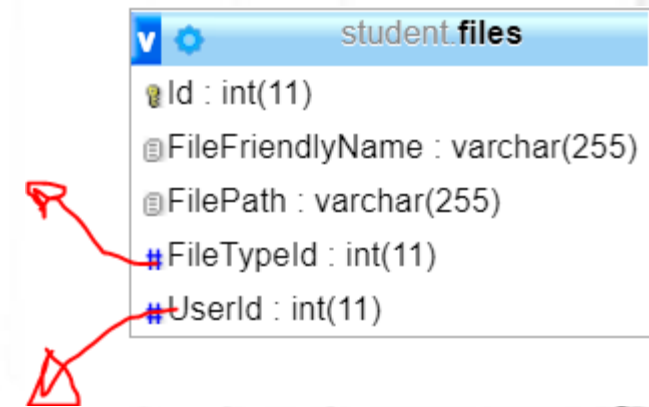
```
echo sha1("Ayman=>Seif 50,Seif=>Mourad 60");  
echo "<br>";  
echo sha1("Ayman=>Seif 5000,Seif=>Mourad 60");
```

```
4365717a0d005e6f07fb8254b3d2ee487e153588  
61d034c3c239cae3cf3edafacde231e6c3c832e2
```

Sha1(Row1,Row2,Row3)  
Sha1("25","25","25")

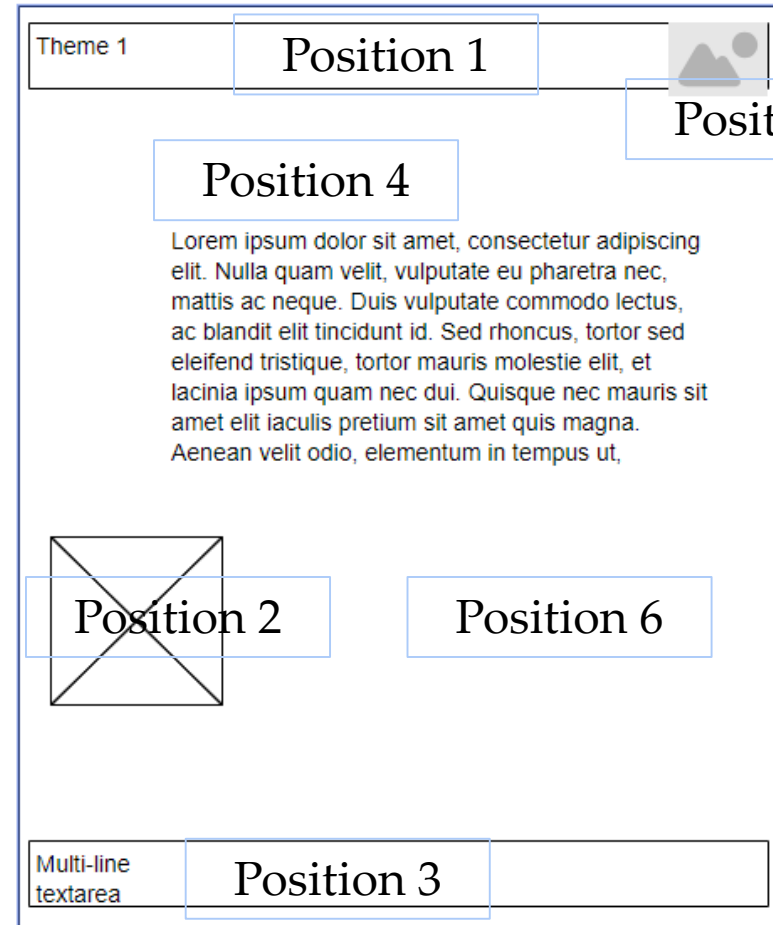
# Patterns in DB -5 Storing Files

- Don't store files as binary in Database (Blob) Platform Dependent.
- Make Size of DB increase dramatically.
- **Solution:**
- Store the path only to your files as Varchar
- Use Hash to give your file names unique identity

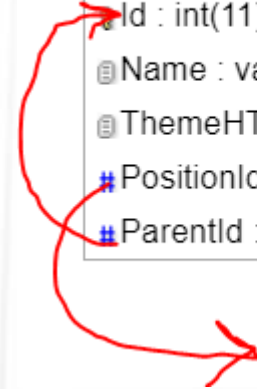


# Patterns in DB -6 Themes and CSS

- Putting Themes in Database with Self reference parent key
- Adding HTML static parts with calling CSS files and even some JS
- Now you can change all the layout of your website dynamically
- Now positions are dynamically allocated



```
student theme
Id : int(11)
Name : varchar(255)
ThemeHTML : text
PositionId : int(11)
ParentId : int(11)
```



# Patterns in DB -7 Emails and Error Messages

- Remember don't Hardcode your Messages

“Nothing under the Double Quotes “

student.tb_users	
ID	: int(11)
FullName	: varchar(255)
Email	: varchar(255)
DOB	: varchar(255)
Password	: varchar(255)
UserType_id	: int(11)
IsDeleted	: tinyint(1)
LastUpdated	: timestamp

student.messageuser	
Id	: int(11)
MessageId	: int(11)
UserId	: int(11)
DateTimeStamp	: timestamp

student.messages	
Id	: int(11)
MessageTemplate	: text
TypeId	: int(11)



عزيزي ايمن،

لقد تعهدنا لك العام الماضي أننا سنصبح أفضل وأسرع وأكبر مما كنا عليه، وبفض  
في تحقيق ذلك. ففي عام ٢٠١٨، تمكنا من زيادة عدد الطلبات إلى أربعة أضعاف  
أننا ضاعفنا عدد المدن و المناطق التي اشتغلنا فيها بثلاثة أضعاف، وعملنا على ج  
اليومية من خلال التوسع في قطاعاتنا الغذائية وغير الغذائية. كذلك قمنا بزيادة  
آلاف في أنحاء دول مجلس التعاون الخليجي،