



جامعة تشرين

كلية الهندسة المعلوماتية

قسم البرمجيات و نظم المعلومات

قواعد معطيات 2

**عنوان المشروع : قاعدة بيانات لشركة سيارات مستعملة**

إعداد : جعفر طالب محمود

إشراف :م. هلا أسعد

## الفهرس:

3	.....	<a href="#">السيناريو</a>	•
4	.....	<a href="#">بنية الجداول</a>	•
5	.....	<a href="#">EER DIAGRAM</a>	•
6	.....	<a href="#">STORED PROCEDURES</a>	•
9	.....	<a href="#">FUNCTIONS</a>	•
10	.....	<a href="#">TRIGGERS</a>	•
11	.....	<a href="#">INDEXES</a>	•
12	.....	<a href="#">INVENTORY</a>	•

## السيناريو

قاعدة المعطيات هي لشركة صغيرة مهتمة بشراء و بيع السيارات المستعملة , لديها عدد من السيارات و الموظفين و العملاء و الموردين.

تحتاج الشركة إلى تتبع عمليات الشراء و البيع و الفحص الخاصة بالسيارات , بالإضافة إلى تتبع معلومات الموظفين و العملاء و الموردين .

تضم قاعدة البيانات ثمانية جداول و هي : السيارات , الموظفين , العملاء , الموردين , المشتريات المبيعات , صور السيارات , عمليات الفحص .

تبدأ عملية الإدخال البيانات مع بداية الشركة , حيث يتم أولاً إضافة الموظفين العاملين في الشركة , و هذا أمر واجب قبل البدء بإدخال أية بيانات أخرى .

في البداية عند حدوث عملية شراء لسيارة معينة , يتم إدخال معلومات البائع في جدول الموردين و بعدها يمكننا إدخال معلومات الشراء في جدول المشتريات.

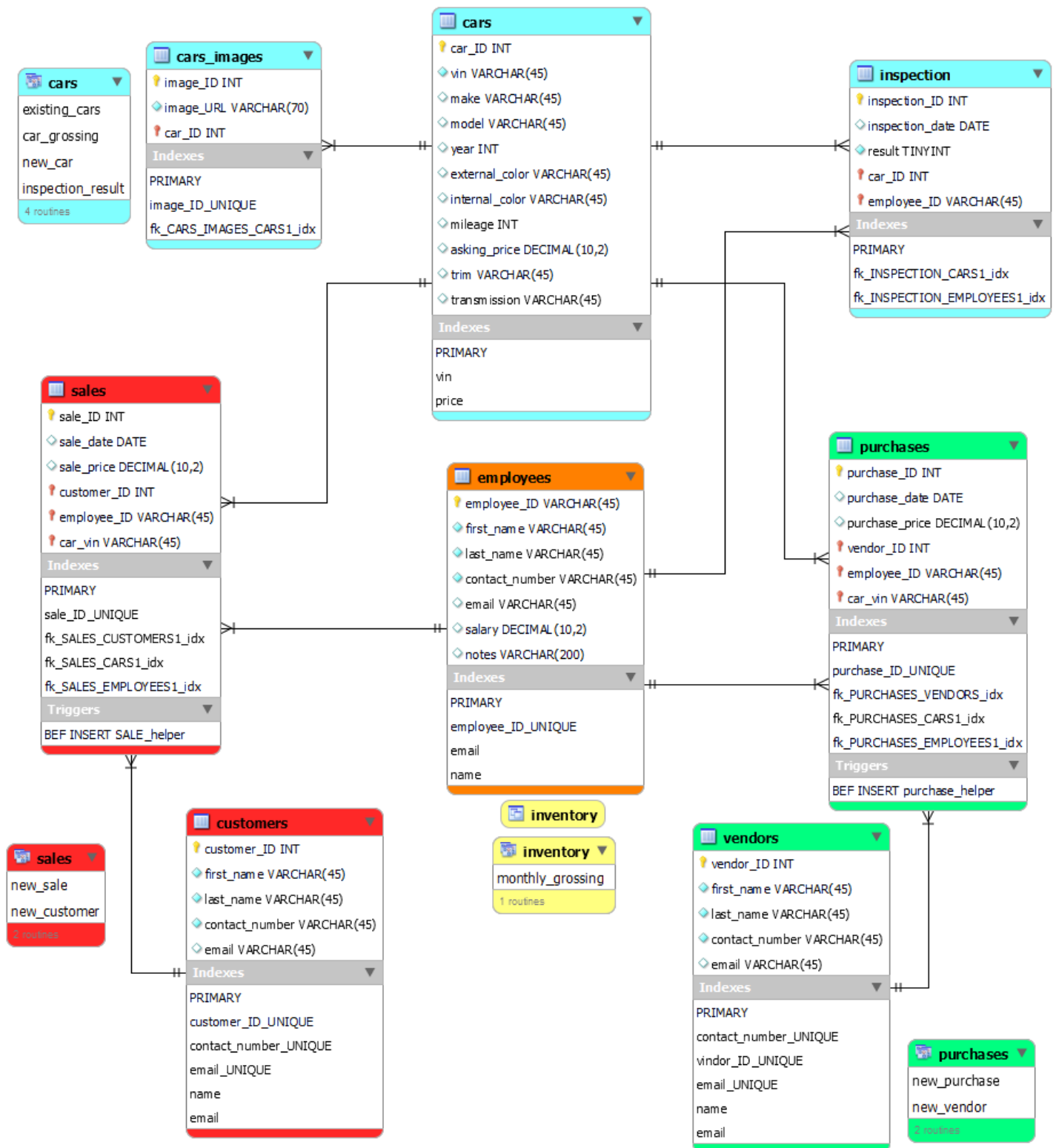
يمكن بعد ذلك إدخال المعلومات التفصيلية للسيارة التي تم شراؤها في جدول السيارات , و كذلك يمكن إدخال صور للسيارة في جدول الصور و إدخال معلومات سجل الفحص الخاص بالسيارة في جدول عمليات الفحص.

فيما بعد في حال تم بيع سيارة معينة , يتم إدخال معلومات العميل في جدول العملاء و بعدها يمكننا إدخال معلومات البيع في جدول المبيعات.

## بنية الجداول

1. الموظفين: لكل موظف رقم تعريف شخصي (مفتاح رئيسي), اسم أول , اسم أخير , رقم هاتف, بريد إلكتروني, راتب , ملاحظات .
2. الموردين: لكل مورد رقم تعريف (مفتاح رئيسي), اسم أول , اسم أخير , رقم هاتف, بريد إلكتروني.
3. العملاء: لكل عميل رقم تعريف (مفتاح رئيسي), اسم أول , اسم أخير , رقم هاتف, بريد إلكتروني.
4. المشتريات : لكل عملية شراء يوجد رقم تعريف (مفتاح رئيسي), تاريخ , سعر , رقم تعريف المورد (مفتاح أجنبي) , بالإضافة إلى رقم تعريف الموظف المشرف عليها (مفتاح أجنبي) , و رقم لوحة المركبة (مفتاح أجنبي).
5. المبيعات : لكل عملية بيع يوجد رقم تعريف (مفتاح رئيسي), تاريخ , سعر , رقم تعريف العميل (مفتاح أجنبي) , بالإضافة إلى رقم تعريف الموظف المشرف عليها (مفتاح أجنبي) , و رقم لوحة المركبة (مفتاح أجنبي).
6. السيارات :لكل سيارة رقم تعريف (مفتاح رئيسي) ,رقم لوحة المركبة, الشركة المصنعة, الموديل , تاريخ الصنع , اللون الخارجي , اللون الداخلي, عدد الأميال المقطوعة , السعر المطلوب بها , التزيين , و نوع ناقل الحركة.
7. سجلات الفحص: لكل فحص رقم تعريف (مفتاح رئيسي) , تاريخ , نتيجة , بالإضافة إلى رقم تعريف الموظف المشرف على الفحص (مفتاح أجنبي) , و رقم تعريف السيارة (مفتاح أجنبي) .
8. صور السيارات : لكل صورة رقم تعرف (مفتاح رئيسي), و مسار تخزين الصورة في نظام الملفات , و رقم تعريف السيارة (مفتاح أجنبي).

# EER DIAGRAM



# STORED PROCEDURES

1. إجراء لإضافة مورد جديد: يراعي حالة الأحرف عند إدخال الاسم و يطبع رسالة تأكيد العملية .

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `new_vendor` (  
2     firstname varchar(45),  
3     lastname varchar(45),  
4     contactnumber varchar(45),  
5     email varchar(45))  
6 BEGIN  
7     insert into vendors (first_name,last_name,contact_number,email)  
8     value(lower(firstname),lower(lastname),lower(contactnumber),lower(email));  
9     select concat(firstname,' ',lastname,' inserted')as result;  
10 END
```

2. إجراء لإضافة عملية شراء جديدة: هنا تم تعريف handler للتعامل مع المشكلة الشائعة (1452) و هي إدخال مفتاح أجنبي في جدول ابن غير موجود في الجدول الأب , أي في حال تم إدخال رقم تعريف مورد غير موجود أو موظف غير موجود , هنا سيتم عرض رسالة بالمشكلة بدلاً من حدوث خطأ يوقف النظام , و في حال تم إدخال البيانات بشكل سليم يقوم بطباعة رسالة تأكيد الإدخال.

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `new_purchase` (purchase_date date,  
2     purchase_price decimal(10,2),vendorID int,  
3     car_vin varchar(45),employeeID varchar(45))  
4 BEGIN  
5 DECLARE exit HANDLER FOR 1452 begin  
6 if employeeID not in(select employee_ID from employees) then  
7     select concat("no such an employee ",employeeID)as result;  
8 elseif vendorID not in (select vendor_ID from vendors)then  
9     select concat("no such a vendor ",vendorID," you have to enter the vindor first")as result;  
10 end if;  
11 end;  
12 insert into purchases(purchase_date,purchase_price,vendor_ID,car_vin,employee_ID)  
13 value(purchase_date,purchase_price,vendorID,upper(car_vin),lower(employeeID));  
14 select concat(car_vin," purchase inserted") as result;  
15 END
```

3. إجراء لإضافة عميل جديد: يراعي حالة الأحرف عند إدخال الاسم و يطبع رسالة تأكيد العملية.

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `new_customer`(  
2     firstname varchar(45),  
3     lastname varchar(45),  
4     contact_number varchar(45),  
5     email varchar(45))  
6 BEGIN  
7     insert into customers (first_name,last_name,contact_number,email)  
8     value(lower(firstname),lower(lastname),lower(contact_number),lower(email));  
9     select concat(firstname,' ',lastname,' inserted')as result;  
10 END
```

4. إجراء لإضافة عملية بيع جديدة: تم تعريف handler للتعامل مع المشكلة (1452) أي في حال تم إدخال رقم تعريف عميل غير موجود أو موظف غير موجود ,هنا سيتم عرض رسالة بالمشكلة بدلاً من حدوث خطأ يوقف النظام ,و في حال تم إدخال البيانات بشكل سليم يقوم بطباعة رسالة تأكيد.

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `new_sale`(saledate date,  
2     sale_price decimal(10,2),customerID int,  
3     carvin varchar(45),employeeID varchar(45))  
4 BEGIN  
5 DECLARE exit HANDLER FOR 1452 begin  
6 if employeeID not in(select employee_ID from employees) then  
7     select concat("no such an employee ",employeeID)as result;  
8 elseif customerID not in (select customer_ID from customers)then  
9     select concat("no such a customer ",customerID," you have to enter the customer first")as result;  
10 end if;  
11 end;  
12 if carvin not in (select car_vin from sales)then  
13     insert into sales(sale_date,sale_price,customer_ID,car_vin,employee_ID)  
14     value(saledate,sale_price,customerID,upper(car_vin),lower(employeeID));  
15     select concat(car_vin," sale inserted") as result;  
16 else select concat('the car ',carvin,' have been soled')as result;  
17 end if;  
18 END
```

5. إجراء لإضافة سيارة جديدة: عند القيام بإضافة عملية شراء جديدة سيقوم القادح [purchase helper](#) بإضافة رقم لوحة المركبة إلى جدول السيارات تلقائياً , بالتالي عند إدخال سيارة يجب التحقق من أنه تم إدخال السيارة بواسطة عملية الشراء أولاً, في هذا الإجراء تم استعمال مبدأ المناقلاات حيث أنه يتم إدخال السيارة في البداية ثم يتم إختبار وجود رقم اللوحة مسبقاً , في حال كان موجوداً يتم تأكيد مناقلة الإدخال و إلا يتم التراجع عنها , و يتم طباعة رسالة بالنتيجة أياً تكن.

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `new_car`(nVIN varchar(45),make varchar(45),
2     model varchar(45),year int,external_color varchar(45),internal_color varchar(45),
3     mileage int,asking_price decimal(10,2),trim varchar(45),transmission varchar(45))
4 BEGIN
5     declare newvin varchar(45) default null;
6     START TRANSACTION;
7     set newvin=upper(nVIN);
8     if upper(nVIN) in (select vin from cars where make is null) then commit ;
9     else rollback;
10    select concat("no such a vin",nvin,"you have to enter the purchase first");
11    end if;
12    START TRANSACTION;
13    if newvin is not null then
14        update cars set make=lower(make),model=lower(model),year=year,external_color=lower(external_color),
15        internal_color=lower(internal_color),mileage=mileage,asking_price=asking_price,trim=lower(trim),
16        transmission=lower(transmission)
17        where vin=newvin;
18        select concat(make," ",model," inserted") as result;commit;
19    else rollback;end if;END
```

6. إجراء يرد السيارات الموجودة حالياً لدى الشركة (أي التي تم شراؤها و لم يتم بيعها بعد):

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `existing_cars`()
2 BEGIN
3     select concat(make,' ',model,' ',year) as car from cars
4     where upper(vin) not in(select car_vin from sales);
5 END
```



# FUNCTIONS

1. تابع يطبع نتيجة آخر فحص تم إجراؤه للسيارة التي تم تمرير رقم تعريفها له، علماً أنه يتم تخزين النتائج كقيم بوليانية في الجدول (1 إذا اجتازت السيارة الفحص و 0 إذا لم تجتازه)

```
1 • CREATE DEFINER=`root`@`localhost` FUNCTION `inspection_passed`(carid int)
2 RETURNS varchar(50) CHARSET utf8mb3 DETERMINISTIC
3 BEGIN
4 DECLARE r boolean;
5     SELECT result INTO r FROM INSPECTION
6         WHERE upper(car_ID)= upper(carid) AND inspection_date =
7             (SELECT MAX(inspection_date) FROM INSPECTION WHERE upper(car_ID)= upper(carid));
8     IF r THEN RETURN "the car passed the latest inspection";
9     ELSE RETURN "the car did not pass the latest inspection" ;
10 END IF;
11 END
```

2. تابع يرد قيمة الربح الناتج عن بيع السيارة التي تم تمرير رقم لوحتها له

```
1 • CREATE DEFINER=`root`@`localhost` FUNCTION `car_grossing`(vin varchar(45)) RETURNS decimal(10,2)
2 DETERMINISTIC
3 BEGIN
4 set @saleprice= (select sale_price from sales where car_vin=upper(vin));
5 set @purchaseprice= (select purchase_price from purchases where car_vin=upper(vin));
6 RETURN @saleprice - @purchaseprice;
7 END
```

3. تابع يرد قيمة الربح الكلي خلال شهر معين يتم تمريره له، تم استخدام مبدأ المؤشرات لتسهيل التعامل مع سجلات من عدة أعمدة و جداول.

```
1 • CREATE DEFINER=`root`@`localhost` FUNCTION `monthly_grossing`( month int) RETURNS decimal(10,2)
2 DETERMINISTIC
3 BEGIN
4 declare flag int default 0; declare mon date;
5 declare PPrice ,SPrice ,grossing decimal(10,2) default 0;
6 DECLARE NOT_FOUND condition for sqlstate value'02000';
7 DECLARE GET_PRICES CURSOR FOR SELECT purchase_price, sale_price, sale_date
8     FROM purchases INNER JOIN sales ON purchases.car_vin = sales.car_vin;
9 DECLARE continue HANDLER FOR NOT_FOUND SET flag = 1;
10
11 open GET_PRICES;
12 while(flag=0)do
13 fetch GET_PRICES INTO PPrice,SPrice,mon;
14 if flag=0 and MONTH(mon) = month then set grossing = grossing+ (SPrice-PPrice);
15 end if;
16 end while;
17 close GET_PRICES;
18 return grossing;
19 END
```

# TRIGGERS

1. قاذح مساعد لإدخال المشتريات : هو قاذح قبل الإدخال في حال عدم إدخال تاريخ لعملية البيع يقوم بإدخال تاريخ اليوم تلقائياً , و كذلك يقوم بإدخال رقم لوحة المركبة إلى جدول السيارات ليعمل هو و إجراء new car على التأكد من أن السيارة تم إدخالها من خلال عملية شراء حصراً.

```
1 • CREATE DEFINER=`root`@`localhost` TRIGGER `purchase_helper`  
2   BEFORE INSERT ON `purchases` FOR EACH ROW  
3 BEGIN  
4 if new.purchase_date is null then  
5   set new.purchase_date =curdate();end if;  
6   insert into `used_cars`.`cars` (vin)value(upper(new.car_vin));  
7  
8   END
```

2. قاذح مساعد لإدخال المبيعات : هو قاذح قبل الإدخال في حال عدم إدخال تاريخ لعملية البيع يقوم بإدخال تاريخ اليوم تلقائياً.

```
1 • CREATE DEFINER=`root`@`localhost` TRIGGER `SALE_helper`  
2   BEFORE INSERT ON `sales` FOR EACH ROW  
3 BEGIN  
4 if new.sale_date is null then  
5   set new.sale_date =curdate();end if;  
6   END
```

# INDEXES

1. فهارس اسماء الموردين و العملاء و الموظفين: وهي فهارس مركبة تبحث في الاسم الأول يليه الاسم الأخير , و فهارس اسماء الموظفين تم تعريفه غير مرئي حالياً.

```
1 • CREATE INDEX name
2   ON vendors(first_name,last_name)comment 'by first name and/or last name';
3 • CREATE INDEX name
4   ON customers(first_name,last_name)comment 'by first name and/or last name';
5 • CREATE INDEX name
6   ON employees(first_name,last_name)invisible comment 'by first name and/or last name';
```

2. فهارس البريد الإلكتروني للموردين و العملاء و الموظفين : بما أن كل بريد إلكتروني ينتهي بنفس المحارف بالتالي يمكن تجاهلها أثناء الفهرسة .

```
1 • CREATE INDEX email
2   ON vendors(email(15)) comment 'no need for @gmail.com';
3 • CREATE INDEX email
4   ON customers(email(15))comment 'no need for @gmail.com';
5 • CREATE INDEX email
6   ON employees(email(15))comment 'no need for @gmail.com';
```

3. فهارس السعر المطلوب: و هو فهارس يبحث تنازلياً .

```
1 • CREATE INDEX PRICE
2   ON CARS(asking_price desc);
```

# INVENTORY VIEW

تم إنشاء منظار خاص بنتائج الجرد , مشتق من جدول السيارات , المبيعات , المشتريات و تم استخدام تابع الربح كذلك , حيث يقوم هذا المنظار بعرض السيارة و تاريخ و سعر الشراء و البيع و الربح الصافي لها .

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `inventory` AS
6     SELECT
7         CONCAT(`c`.`make`,` `,`c`.`model`,` `,`c`.`year`,`:`) AS `Car`,
8         `s`.`sale_date` AS `Sale Date`,
9         `s`.`sale_price` AS `Sale Price`,
10        `p`.`purchase_date` AS `Purchase Date`,
11        `p`.`purchase_price` AS `Purchase Price`,
12        CAR_GROSSING(`c`.`vin`) AS `Grossing`
13    FROM
14        ((`purchases` `p`
15        LEFT JOIN `cars` `c` ON ((`c`.`vin` = `p`.`car_vin`)))
16        LEFT JOIN `sales` `s` ON ((`c`.`vin` = `s`.`car_vin`)))
17    ORDER BY `s`.`sale_date` DESC
```

- انتهى -