|  |  |
| --- | --- |
|  | **2016** |
|  | جامعة دمشق  كلية الهندسة المعلوماتية  قسم هندسة البرمجيات ونظم المعلومات |

|  |
| --- |
| **[وظيفة قواعد المعطيات المتقدمة]** |
| إعداد الطلاب: آلاء الحموي محمد باسل الشمالي محمد غانم هانئة المالكي |

# 1. أوراكل:

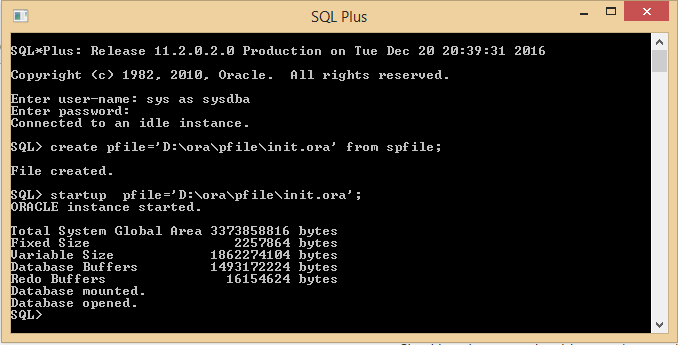
## الطلب الأول:

لإنشاء pfile نستخدم التعليمة التالية:

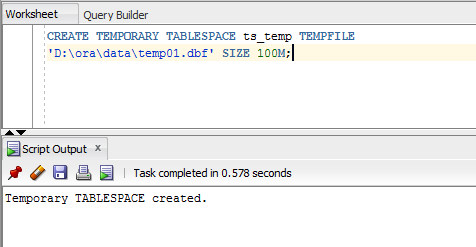
create pfile='D:\ora\pfile\init.ora' from spfile;

الآن نقلع قاعدة البيانات من الـ pfile الذي تم إنشاءه:

STARTUP pfile 'D:\ora\pfile\init.ora';



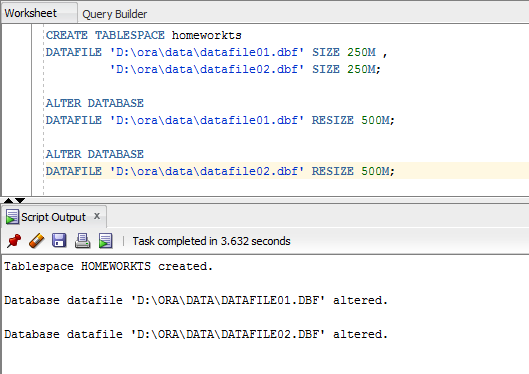
## الطلب الثاني:



## الطلب الثالث:

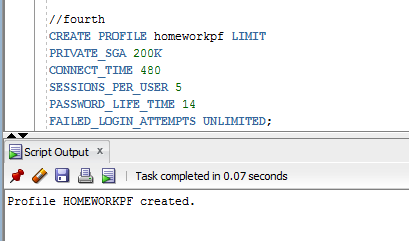
بدايةً، نقوم بإنشاء tablespace حجمه 500 Mega وذلك بجعله مكون من ملفين كل منهما حجمه 250 Mega

بعد ذلك نقوم بزيادة حجمه وذلك بزيادة حجم الملفات التي تكونه

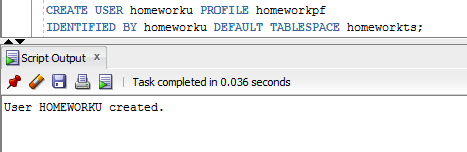


## الطلب الرابع:

إنشاء profile وجعل الذاكرة200 KB ومدة الاتصال 8 ساعات أي480 دقيقة وعدد الجلسات التي يستطيع نفس المستخدم فتحها 5 وعدد الأسابيع التي يجب بعد مرورها تغيير كلمة السر 2 أي 14 يوم وعدد المرات التي يستطيع إعادة محاولة تسجيل الدخول بعد فشلها غير محدود.

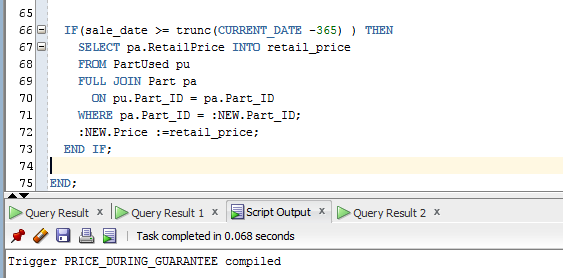
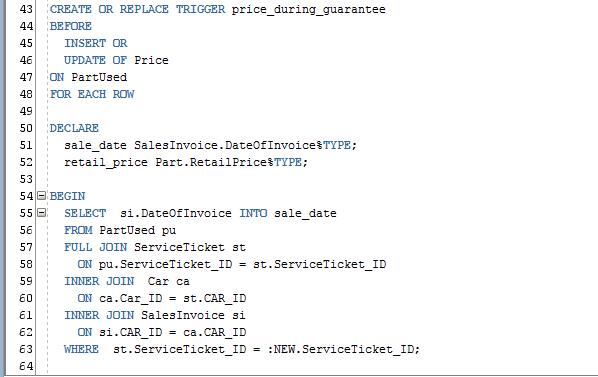


## الطلب الخامس:

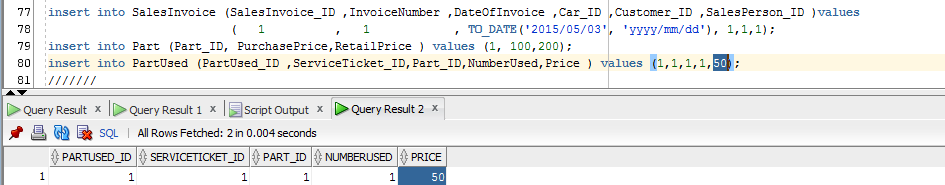


# 2.قواعد المعطيات الفعالة و PL\_SQL:

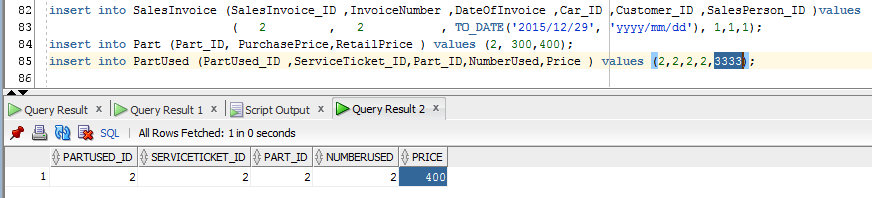
## الطلب الأول:



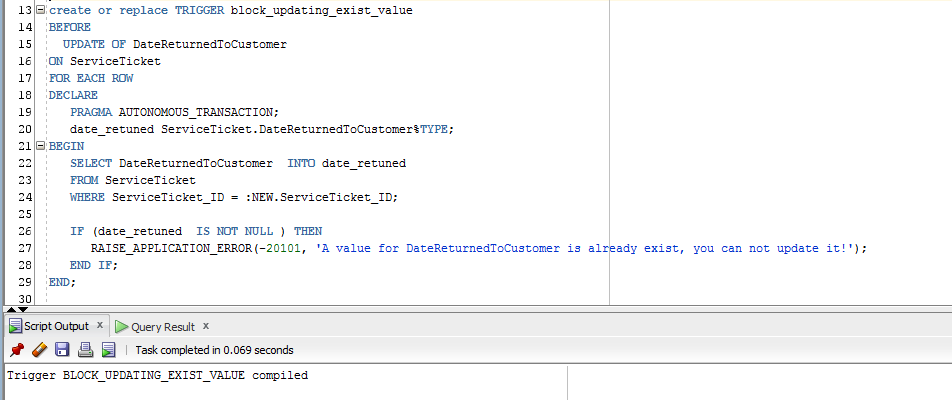
الآن نلاحظ عند إدراج PartUsed تابع لسيارة خارج مدة الكفالة، ييتم تخزين في قاعدة البيانات القيمة المدخلة:



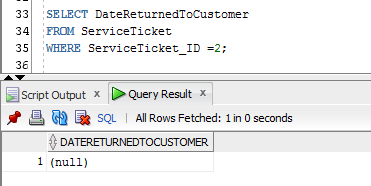
أما عند إدراج PartUsed تابع لسيارة ضمن مدة الكفالة، يتم تخزين قيمة الـ RetailPrice في حقل الـ Price في قاعدة البيانات بدل من القيمة المدخلة:



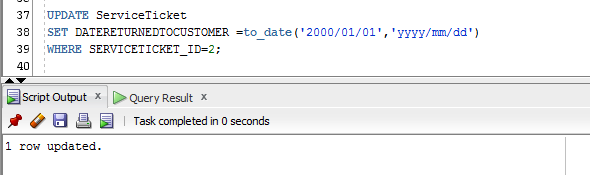
## الطلب الثاني:



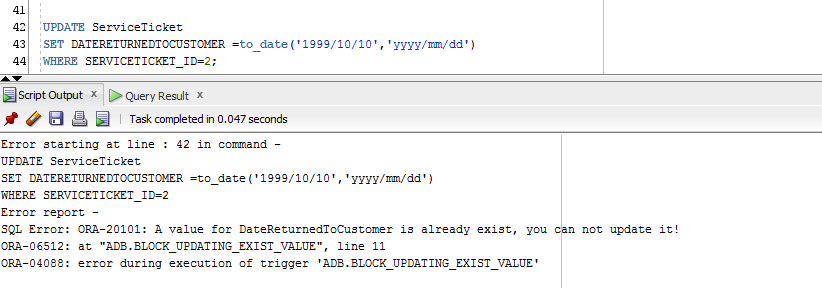
في حال كان لدينا ريكورد لا يملك قيمة:



عند محاولة تعديل هذه القيمة تنجح العملية:



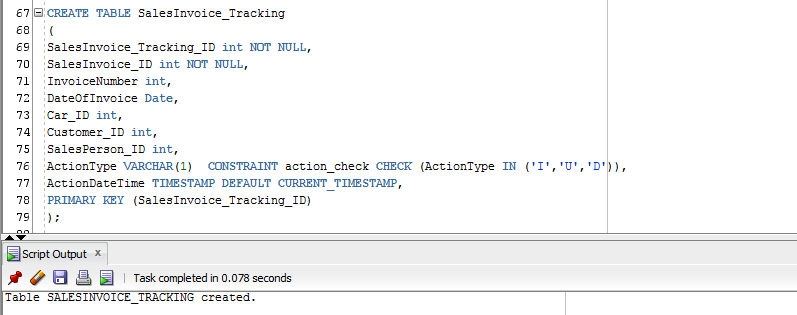
بعد ذلك نحاول تعديل نفس القيمة بعد امتلاكها قيمة سابقة فـ **لا** تتم عملية التعديل بنجاح ويظهر خطأ يمنع إتمام العملية:



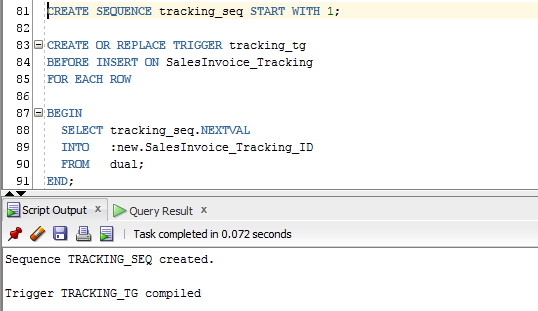
## الطلب الثالث:

## الطلب الرابع:

يتم بناء جدول SalesInvoice\_Tracking تحفظ فيه جميع القديمة الموجودة في جدول SalesInvoice قبل تعديلها، مع ذكر تاريخ وزمن التعديل:

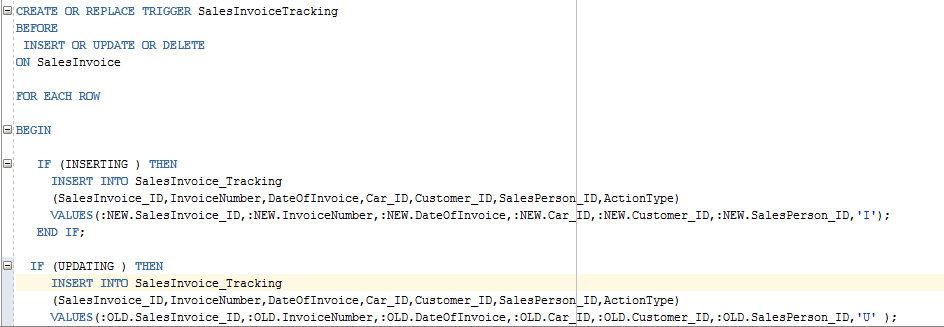


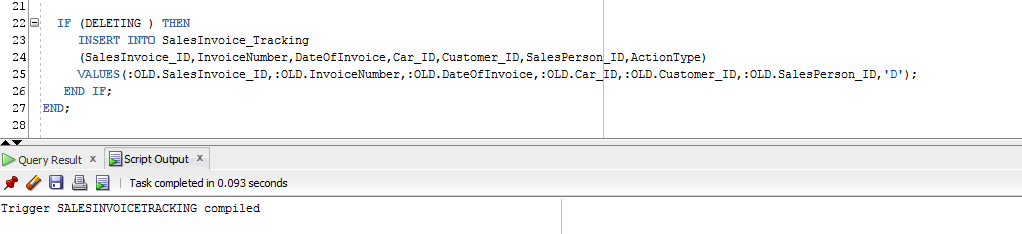
وبما أن النسخة 11 من أوراكل لا تدعم الـ IDENTITY يتم استخدام SEQUENCE للحصول على ID متسلسل أوتوماتيكي وبناء قادح يقدح قبل أي عملية ادخال تتم على الجدول



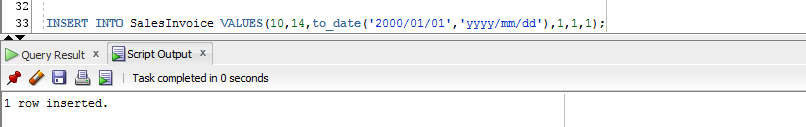
ملاحظة: تم إضافة ميزة الـ IDENTITY في النسخة 12

بعد ذلك يتم بناء قادح على جدول فاتورة البيع SalesInvoice، ويُقدح قبل البدء بأي عملية DML، إذ يقوم بتخزين القيم القديمة في الجدول SalesInvoice\_Tracking مع إضافة تاريخ وزمن التعديل:

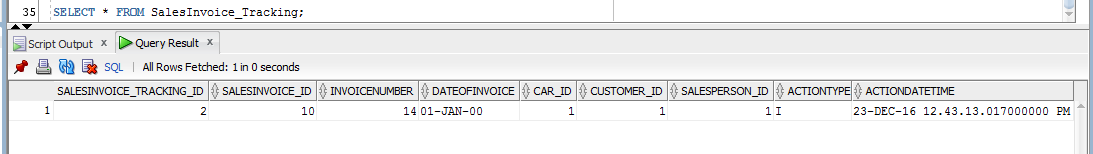


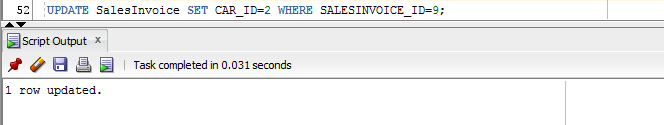


الآن عند القيام بأي عملية إضافة – تعديل – حذف على جدول الفواتير ينعكس ذلك على جدول التتبع:

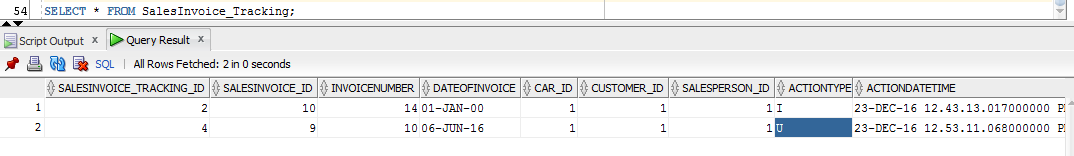


نلاحظ أن القيمة انعكست على جدول المتابعة:



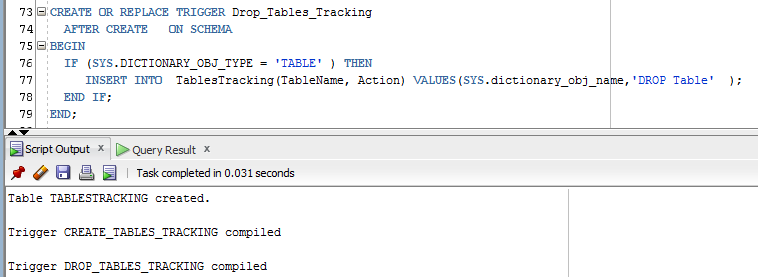
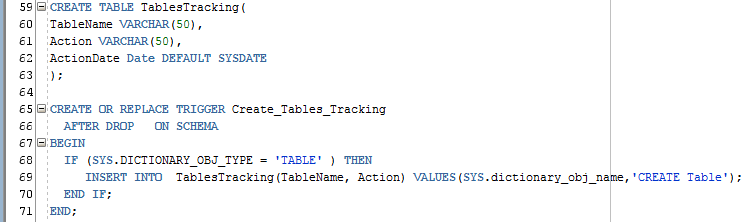
وكذلك الأمر بالنسبة لعملية الـ Update

نلاحظ أن القيمة قبل التعديل قد تم ادراجها في سجل المتابعة:

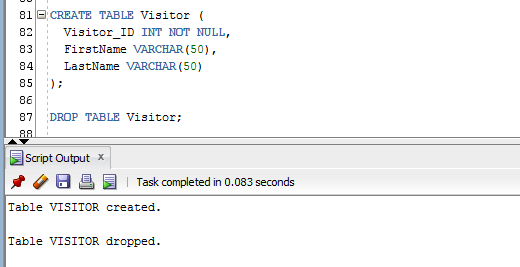


## الطلب الخامس:

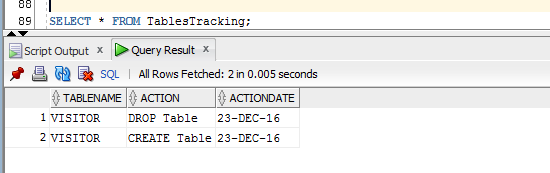
نقوم ببناء الجدول TablesTracking الذي سيضم معلومات عن الجداول التي يتم إنشاءها وحذفها، ثم قوادح من نمط DDL تنقدح في حال القيام بعملية إنشاء أو حذف جدول وتخزن العملية في الجدول TablesTracking



نقوم بعملية إضافة وحذف لنفس الجدول



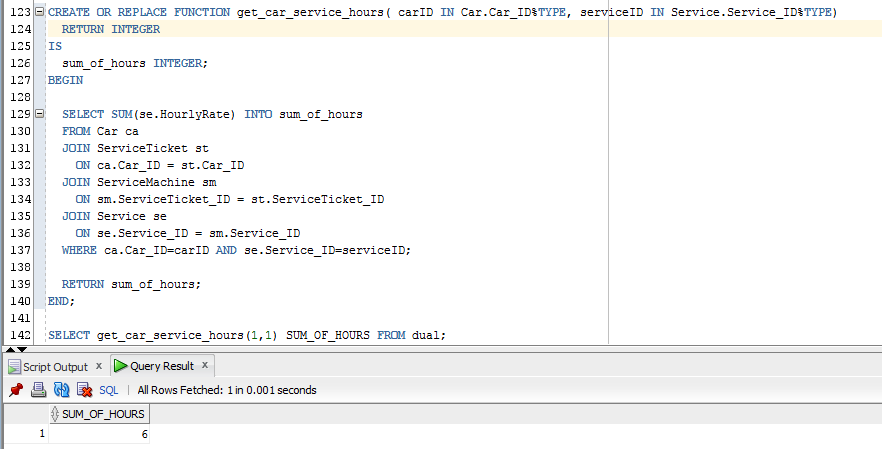
نلاحظ أن نتيجة الاضافة والحدث قد تم الاحتفاظ بها

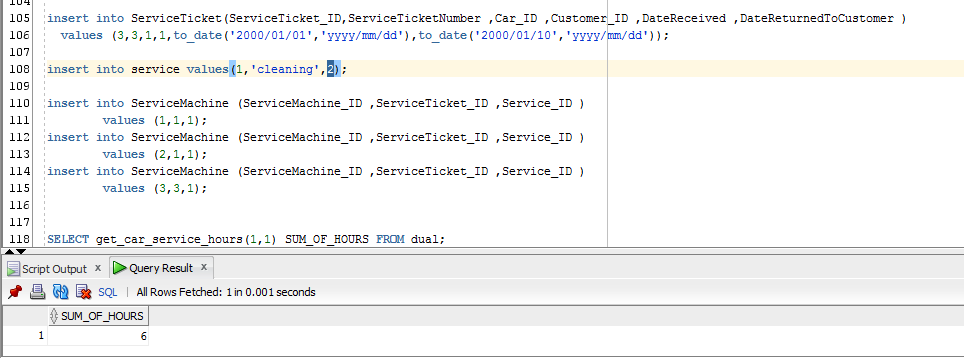


## الطلب السادس:

نقوم بخلق تابع يستقبل رقم الخدمة والسيارة ويعيد عدد الساعات المقضية على هذه الخدمة للسيارة المطلوبة

مع ملاحظة أنه في مثالنا عدد الساعات المطلوبة للخدمة 1 هي ساعتين وعدد المرات التي طلبت بها هذه الخدمة ثلاث مرات.





# 3.قواعد المعطيات متعددة الأبعاد:

## الطلب الأول: