**حل مسألة الوظيفة: TIC\_IIS303**

**مدرس المقرر:** الدكتور فادي غسان إبراهيم

**الفصل الدراسي:** F24

**أولاً: بناء قاعدتي البيانات مع الجداول (المطلوب ١)**

فيما يلي أوامر SQL لإنشاء قاعدتي البيانات والجداول المذكورة في المسألة.

**إنشاء قاعدة بيانات المستشفى (hospital\_db) وجداولها:**

-- إنشاء قاعدة بيانات المستشفى

CREATE DATABASE hospital\_db CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

-- استخدام قاعدة البيانات

USE hospital\_db;

-- إنشاء جدول المرضى

CREATE TABLE patient (

    patient\_id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(255) NOT NULL,

    phone VARCHAR(20),

    insurance\_id VARCHAR(100)

);

-- إنشاء جدول الأطباء في المستشفى

CREATE TABLE doctor (

    doctor\_id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(255) NOT NULL,

    specialty VARCHAR(100),

    password VARCHAR(255) NOT NULL,

    last\_password\_change TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- إنشاء جدول الزيارات

CREATE TABLE appointment (

    appointment\_id INT PRIMARY KEY AUTO\_INCREMENT,

    patient\_id INT,

    doctor\_id INT,

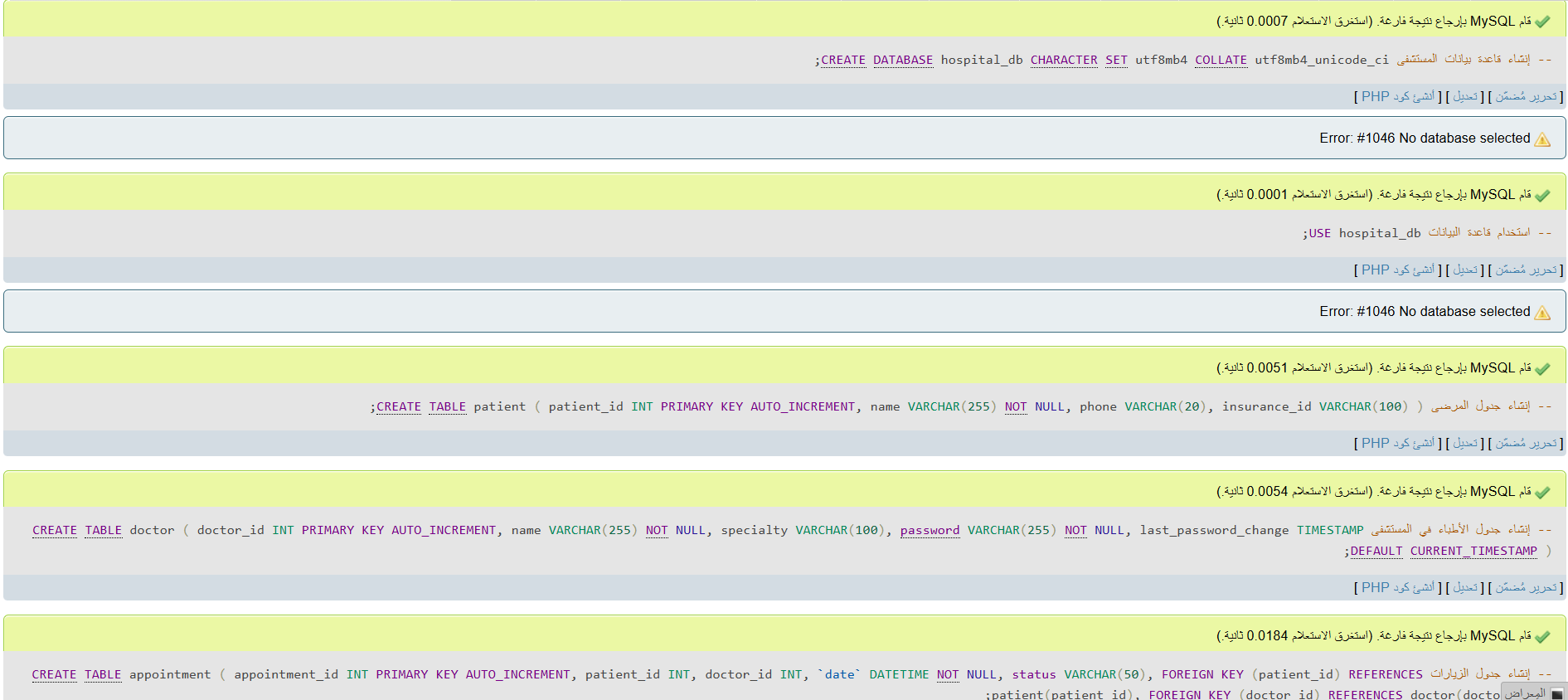
    `date` DATETIME NOT NULL,

    status VARCHAR(50),

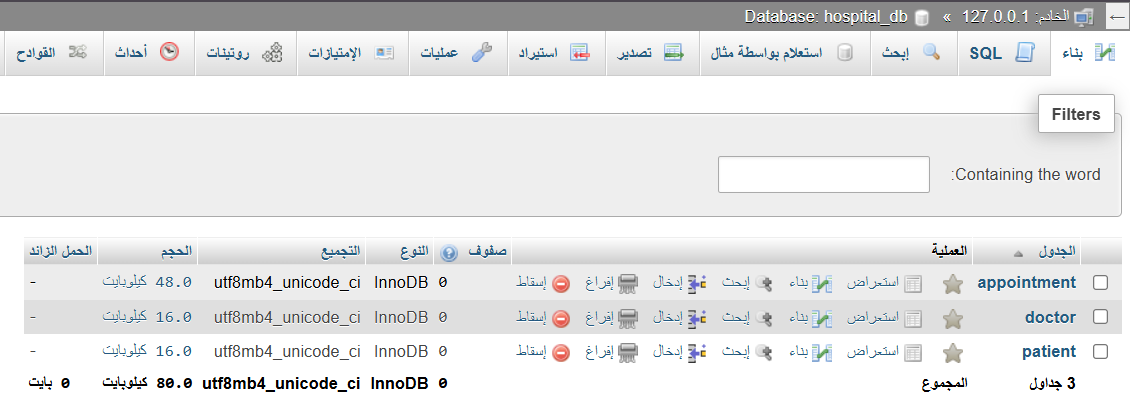
    FOREIGN KEY (patient\_id) REFERENCES patient(patient\_id),

    FOREIGN KEY (doctor\_id) REFERENCES doctor(doctor\_id)

);



تنفيذ الطلب



تم الانشاء

**2. إنشاء قاعدة بيانات شركة التأمين (insurance\_db) وجداولها:**

-- إنشاء قاعدة بيانات شركة التأمين

CREATE DATABASE insurance\_db CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

-- استخدام قاعدة البيانات

USE insurance\_db;

-- إنشاء جدول العملاء

CREATE TABLE client (

    client\_id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(255) NOT NULL,

    policy\_number VARCHAR(100) UNIQUE,

    password VARCHAR(255) NOT NULL,

    last\_password\_change TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- إنشاء جدول الأطباء في شركة التأمين

CREATE TABLE doctor (

    doctor\_id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(255) NOT NULL,

    specialty VARCHAR(100),

    password VARCHAR(255) NOT NULL,

    last\_password\_change TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    dual\_password VARCHAR(255)

);

-- إنشاء جدول المطالبات

CREATE TABLE claim (

    claim\_id INT PRIMARY KEY AUTO\_INCREMENT,

    client\_id INT,

    doctor\_id INT,

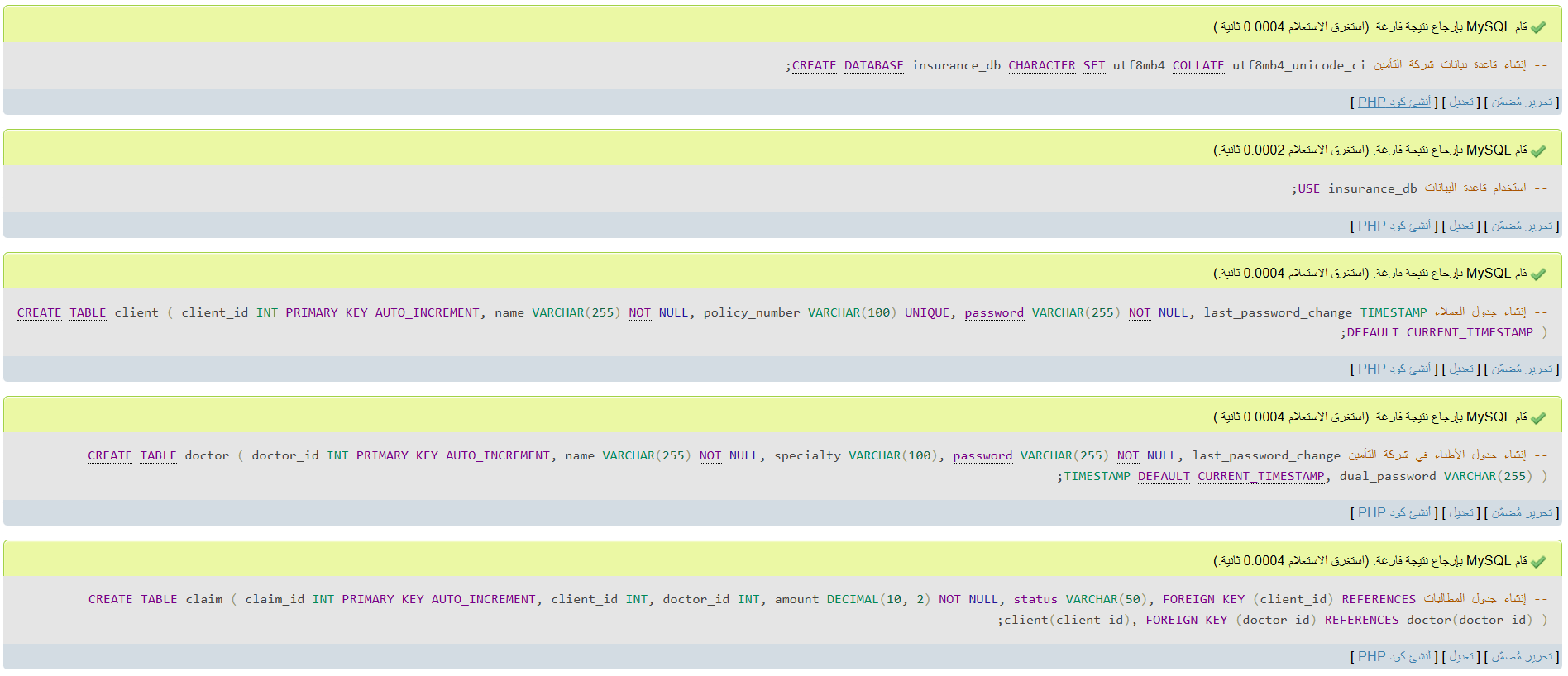
    amount DECIMAL(10, 2) NOT NULL,

    status VARCHAR(50),

    FOREIGN KEY (client\_id) REFERENCES client(client\_id),

    FOREIGN KEY (doctor\_id) REFERENCES doctor(doctor\_id)

);





تم الانشاء

**ثانياً: إنشاء المستخدمين ومنح الصلاحيات (المطلوب ٢)**

-- a. إنشاء مدير النظام ومنحه الصلاحيات الكاملة

CREATE USER 'SysAdmin'@'localhost' IDENTIFIED BY 'Sys@dmin\_12345678';

GRANT ALL PRIVILEGES ON \*.\* TO 'SysAdmin'@'localhost' WITH GRANT OPTION;

-- b. إنشاء مسؤول المستشفى ومنحه الصلاحيات الكاملة على قاعدة بيانات المستشفى

CREATE USER 'HospitalAdmin Ayla'@'localhost' IDENTIFIED BY 'Admin\_12345';

GRANT ALL PRIVILEGES ON hospital\_db.\* TO 'HospitalAdmin Ayla'@'localhost';

-- c. إنشاء مسؤول التأمين ومنحه الصلاحيات الكاملة على قاعدة بيانات شركة التأمين

CREATE USER 'InsuranceAdmin Reema'@'localhost' IDENTIFIED BY 'Insurance\_123';

GRANT ALL PRIVILEGES ON insurance\_db.\* TO 'InsuranceAdmin Reema'@'localhost';

-- d. إنشاء مستخدم الطبيب ومنحه الصلاحيات المحددة

CREATE USER 'Doctor Aram'@'localhost' IDENTIFIED BY 'Doctor\_12345';

-- i. صلاحيات على جداول الأطباء في القاعدتين

GRANT SELECT, INSERT, UPDATE ON hospital\_db.doctor TO 'Doctor Aram'@'localhost';

GRANT SELECT, INSERT, UPDATE ON insurance\_db.doctor TO 'Doctor Aram'@'localhost';

-- ii. صلاحيات على جدول الزيارات

GRANT SELECT, INSERT, UPDATE ON hospital\_db.appointment TO 'Doctor Aram'@'localhost';

-- iii. صلاحية القراءة من جدول المرضى

GRANT SELECT ON hospital\_db.patient TO 'Doctor Aram'@'localhost';

-- iv. صلاحية القراءة من جدول العملاء

GRANT SELECT ON insurance\_db.client TO 'Doctor Aram'@'localhost';

-- v. صلاحيات على جدول المطالبة

GRANT SELECT, INSERT ON insurance\_db.claim TO 'Doctor Aram'@'localhost';

-- e. إنشاء مستخدم العميل ومنحه الصلاحيات المحددة

CREATE USER 'Client Iliam'@'localhost' IDENTIFIED BY 'Client\_123456';

GRANT SELECT ON insurance\_db.client TO 'Client Iliam'@'localhost';

GRANT SELECT, UPDATE ON insurance\_db.claim TO 'Client Iliam'@'localhost';

-- f. إنشاء موظف الاستعلامات ومنحه صلاحيات القراءة فقط

CREATE USER 'Viewer Fady'@'localhost' IDENTIFIED BY 'Viewer\_123456';

GRANT SELECT ON hospital\_db.\* TO 'Viewer Fady'@'localhost';

GRANT SELECT ON insurance\_db.\* TO 'Viewer Fady'@'localhost';

-- تطبيق الصلاحيات

FLUSH PRIVILEGES;

**ثالثاً إلى عاشراً: تنفيذ السياسات والتعديلات المطلوبة**

**٣. سحب امتياز التحديث على جدول المطالبة من العميل Client Iliam:**

REVOKE UPDATE ON insurance\_db.claim FROM 'Client Iliam'@'localhost';

**٤. تعيين سياسة عامة لتغيير كلمة المرور كل 90 يوماً:**

-- هذا الأمر يضبط السياسة لجميع المستخدمين الذين لم يتم تحديد سياسة خاصة لهم

SET PERSIST default\_password\_lifetime = 90;

**٥. تعديل سياسة تغيير كلمة المرور لمسؤول التأمين InsuranceAdmin Reema لتصبح 60 يوماً:**

ALTER USER 'InsuranceAdmin Reema'@'localhost' PASSWORD EXPIRE INTERVAL 60 DAY;

**٦. إلغاء تفعيل تغيير كلمة المرور للمريض:**  
ملاحظة: "المريض" هو سجل في جدول وليس مستخدماً للنظام، لذلك لا يمتلك كلمة مرور. لو كان القصد هو إلغاء انتهاء صلاحية كلمة المرور لمستخدم معين (على سبيل المثال Client Iliam)، فسيكون الأمر كالتالي:

-- مثال افتراضي لإلغاء انتهاء صلاحية كلمة المرور لمستخدم

ALTER USER 'Client Iliam'@'localhost' PASSWORD EXPIRE NEVER;

**٧. تغيير كلمة مرور الطبيب Doctor Aram باستخدام تقنية كلمتي المرور (Dual Password):**  
تسمح هذه التقنية بالانتقال السلس لكلمة المرور في التطبيقات دون انقطاع.

ALTER USER 'Doctor Aram'@'localhost'

IDENTIFIED BY 'Aram\_NewPass2025' RETAIN CURRENT PASSWORD;

-- الخطوة الثانية (بعد تحديث جميع التطبيقات المتصلة لاستخدام كلمة المرور الجديدة): تجاهل كلمة المرور القديمة

ALTER USER 'Doctor Aram'@'localhost' DISCARD OLD PASSWORD;

**٨. إنهاء مدة صلاحية كلمة المرور للعميل Client Iliam (إجباره على التغيير عند الدخول التالي):**

ALTER USER 'Client Iliam'@'localhost' PASSWORD EXPIRE;

**٩. تغيير كلمة مرور موظف الاستعلامات Viewer Fady:**

ALTER USER 'Viewer Fady'@'localhost' IDENTIFIED BY 'F@dy\_1234';

**١٠. اقتراح سياستي نسخ احتياطي:**

لضمان حماية البيانات من الفقدان، يُقترح اعتماد استراتيجية نسخ احتياطي متكاملة تجمع بين نوعين من النسخ الاحتياطي:

**السياسة الأولى: النسخ الاحتياطي المنطقي (Logical Backup) باستخدام mysqldump**

* **الوصف:** تقوم هذه الأداة بإنشاء ملف نصي يحتوي على أوامر SQL اللازمة لإعادة بناء قواعد البيانات والجداول والبيانات.
* **المزايا:**
  + **مرونة عالية:** يمكن استعادة نسخة احتياطية على إصدارات مختلفة من MySQL أو حتى على أنظمة قواعد بيانات أخرى.
  + **حجم أصغر:** عادة ما تكون الملفات مضغوطة وأصغر من النسخ الفيزيائية.
  + **سهولة الاستخدام:** الأوامر بسيطة ويمكن جدولتها بسهولة.
* **خطة التنفيذ المقترحة:**
  + **نسخة احتياطية كاملة يومياً:** تتم جدولتها في وقت متأخر من الليل (فترة انخفاض الاستخدام) للاحتفاظ بنسخة كاملة من البيانات لكل يوم.

**السياسة الثانية: النسخ الاحتياطي الفيزيائي (Physical Backup)**

* **الوصف:** يتم فيها نسخ ملفات البيانات الفعلية التي يستخدمها MySQL مباشرة. يمكن استخدام أدوات مثل Percona XtraBackup للقيام بذلك دون إيقاف الخادم.
* **المزايا:**
  + **سرعة الاستعادة:** عملية الاستعادة أسرع بكثير من النسخ المنطقي، خاصة مع قواعد البيانات الضخمة.
  + **كفاءة:** أقل تأثيراً على أداء الخادم أثناء عملية النسخ.
* **خطة التنفيذ المقترحة:**
  + **نسخة احتياطية كاملة أسبوعياً:** تؤخذ في عطلة نهاية الأسبوع.
  + **نسخة احتياطية تزايدية (Incremental) يومياً:** تقوم بنسخ التغييرات التي حدثت منذ آخر نسخة احتياطية (كاملة أو تزايدية)، مما يوفر مساحة تخزين ووقتاً.

**الاستراتيجية الموصى بها:**  
الجمع بين السياستين لتحقيق أفضل حماية:

1. **نسخ احتياطي فيزيائي كامل كل يوم أحد.**
2. **نسخ احتياطي فيزيائي تزايدي من الاثنين إلى السبت.**
3. **نسخ احتياطي منطقي كامل لكل قاعدة بيانات على حدة يومياً** للاستفادة من مرونته عند الحاجة لاستعادة جدول معين أو بيانات محددة.
4. **تخزين نسخ احتياطية متعددة في موقع جغرافي مختلف** للحماية من الكوارث الطبيعية.