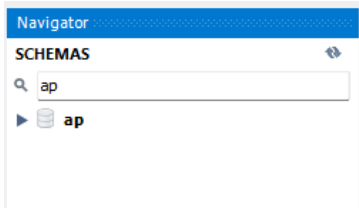
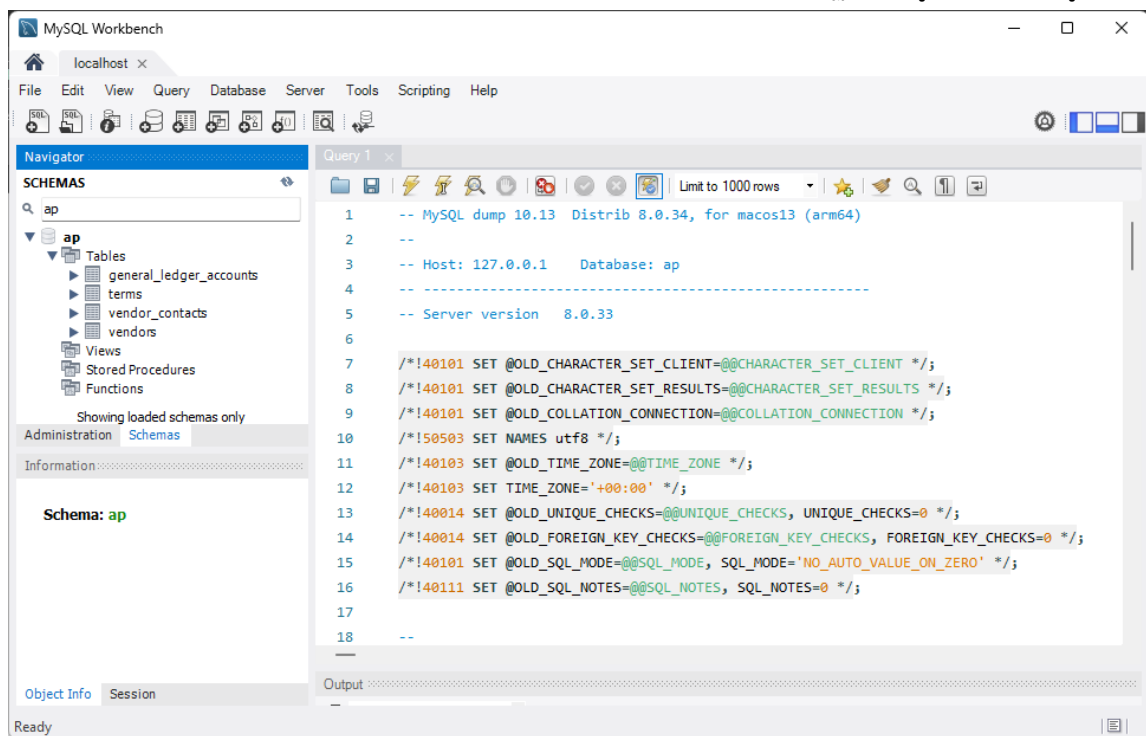


## LAB 7: ClassRelationship

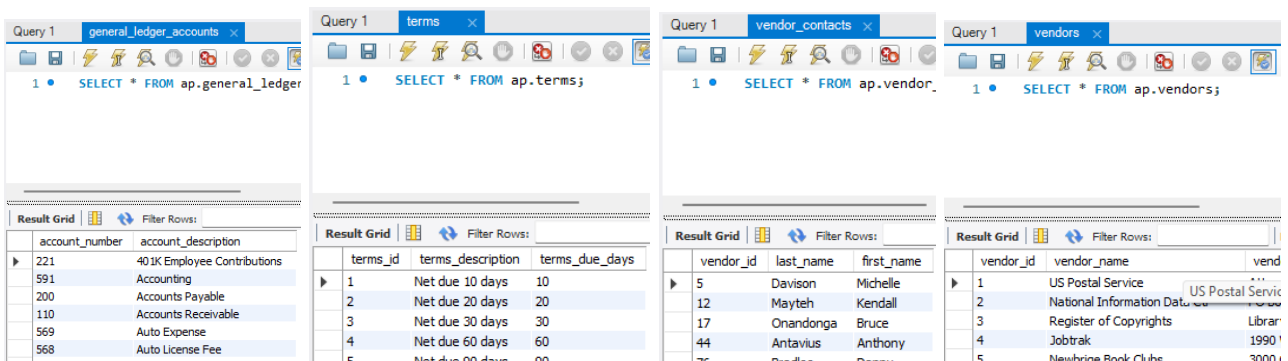
1. จงสร้างฐานข้อมูลชื่อ “ap” ด้วยโปรแกรม “MySQL Workbench” ดังแสดงต่อไปนี้



1.1) ให้นักศึกษาเปิดไฟล์ <https://itsci.mju.ac.th/downloads/watcharin/it214/ap.sql.txt> จากนั้นคัดลอกโค้ด SQL Commands ไปกรอกใน Query Tab จากนั้นรันคำสั่ง SQL เพื่อสร้างตารางฐานข้อมูลดังแสดงดังรูปต่อไปนี้

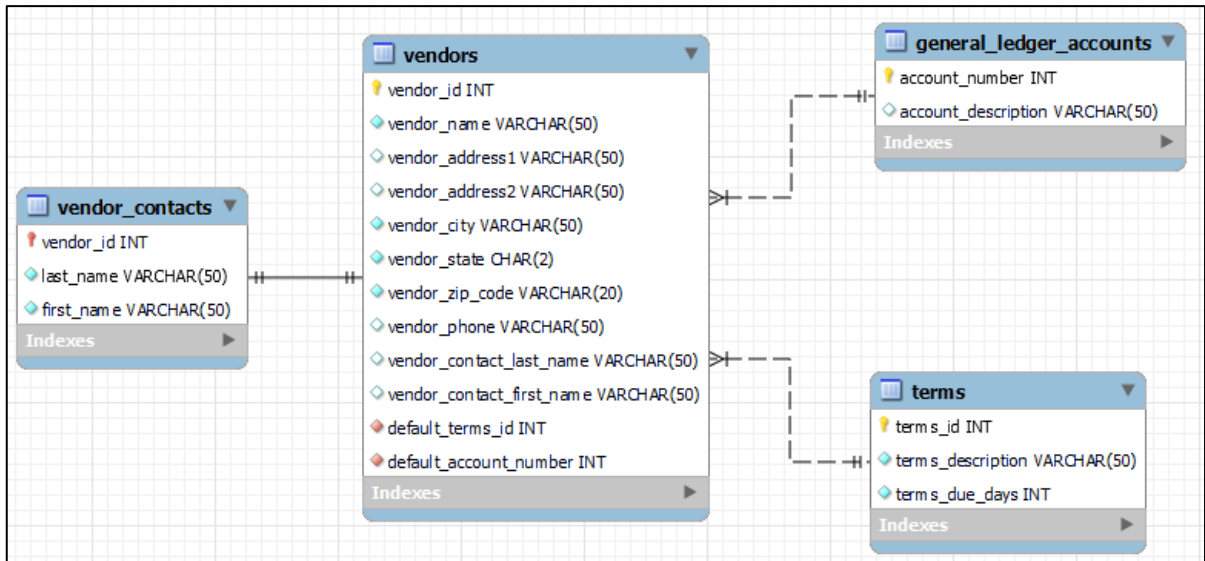


เมื่อรันคำสั่งเสร็จแล้วจะปรากฏตารางและข้อมูลดังแสดง



2. จากฐานข้อมูล “ap” สามารถแสดงเป็น ER Diagram ดังแสดงต่อไปนี้ จงแปลงฐานข้อมูลที่กำหนดให้เป็น Hibernate Class (นักศึกษาสามารถสร้างโปรเจ็คจากตัวอย่างโปรเจ็คได้จากลิงค์นี้

<https://itsci.mju.ac.th/~watcharin/downloads/it214/lab-template/> )



2.1) จงสร้างคลาส Term จากโค้ดที่กำหนดให้เป็น Entity คลาสโดยใช้ Hibernate Annotations จากนั้นแก้ไขโค้ดในไฟล์ TestTermApp.java โดยใช้ Hibernate Query เพื่อแสดงผลลัพธ์ดังแสดงต่อไปนี้

```

--- List of Terms ---
-----
Term ID | Description          | Due Days
-----
1       | Net due 10 days     | 10
2       | Net due 20 days     | 20
3       | Net due 30 days     | 30
4       | Net due 60 days     | 60
5       | Net due 90 days     | 90
-----
    
```

### โค้ด Term.java (ให้นักศึกษาเขียนโค้ดต่อไปนี่)

```
package /* your package */;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "terms")
public class Term {
    @Id
    @Column(name = "terms_id")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer termsId;
    @Column(name = "terms_description", nullable = false, length = 50)
    private String termsDescription;
    @Column(name = "terms_due_days")
    private Integer termsDueDays;

    // TODO: create Getter/Setter functions
    // ...
}
```

### โค้ด TestApp.java ให้นักศึกษาเขียนโค้ดต่อไปนี่พร้อมเติมโค้ดที่จำเป็นในส่วนที่ขาดไป (โค้ดของ HibernateConnection.java แสดงอยู่ด้านล่างสุดของเอกสาร)

```
package /* your package */;

import java.util.List;
import org.hibernate.query.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

public class TestTermApp {
    public static void main(String[] args) {
        SessionFactory sessionFactory = HibernateConnection.getSessionFactory();
        Session session = sessionFactory.openSession();

        try {
            session.beginTransaction();
            String hql = "FROM Term";
            Query<Term> query = session.createQuery(hql, Term.class);
            List<Term> terms = query.getResultList();

            // TODO: Your code to display output here
            // ...

            session.getTransaction().commit();
        } finally {
            if (session.getTransaction().isActive()) {
                session.getTransaction().rollback();
            }
            session.close();
        }
    }
}
```

```
}  
}
```

2.2) จงสร้างคลาส GeneralLedgerAccount ให้เป็น Entity คลาสโดยใช้ Hibernate Annotations จากนั้นสร้างคลาส TestGeneralLedgerAccountApp เพื่อแสดงผลลัพธ์ โดยใช้ Hibernate Query ดังแสดงต่อไปนี้

```
--- List of GeneralLedger Account ---  
-----  
Account Number      | Description  
-----  
221                  | 401K Employee Contributions  
591                  | Accounting  
200                  | Accounts Payable  
110                  | Accounts Receivable  
569                  | Auto Expense  
568                  | Auto License Fee  
565                  | Bank Fees  
394                  | Book Club Royalties  
181                  | Book Development  
120                  | Book Inventory  
-----
```

หมายเหตุ: ใช้คำสั่งดังต่อไปนี้เพื่อจำกัดจำนวน records ในการแสดงผลในช่วง 0 – 10 records แรก

```
...  
String hql = "FROM GeneralLedgerAccount";  
Query<GeneralLedgerAccount> query = session.createQuery(hql, GeneralLedgerAccount.class);  
query.setFirstResult(0);  
query.setMaxResults(10);  
...
```

2.3) จงสร้างคลาส Vendor, VendorContract ให้เป็น Entity โดยใช้ Hibernate Annotation พร้อมกำหนดความสัมพันธ์ระหว่างคลาสต่าง ๆ ตาม ER Diagram ที่กำหนดให้ จากนั้นสร้างคลาส TestVendorApp โดยใช้ Hibernate Query เพื่อแสดงผลพร้อมดังแสดงต่อไปนี้

```

-----
Vendor ID | Name | Address
-----
1 | US Postal Service | Attn: Supt. Window Services
--> Account: { number=552, description=Postage }
--> Term: { id=1, description=Net due 10 days, due days=10 }
2 | National Information Data Ctr | PO Box 96621
--> Account: { number=540, description=Direct Mail Advertising }
--> Term: { id=3, description=Net due 30 days, due days=30 }
3 | Register of Copyrights | Library Of Congress
--> Account: { number=403, description=Book Production Costs }
--> Term: { id=3, description=Net due 30 days, due days=30 }
4 | Jobtrak | 1990 Westwood Blvd Ste 260
--> Account: { number=572, description=Books, Dues, and Subscriptions }
--> Term: { id=3, description=Net due 30 days, due days=30 }
5 | Michelle Davison | 3000 Cindel Drive
--> Account: { number=394, description=Book Club Royalties }
--> Term: { id=4, description=Net due 60 days, due days=60 }
6 | California Chamber Of Commerce | 3255 Ramos Cir
--> Account: { number=572, description=Books, Dues, and Subscriptions }
--> Term: { id=3, description=Net due 30 days, due days=30 }
7 | Towne Advertiser's Mailing Svcs | Kevin Minder
--> Account: { number=540, description=Direct Mail Advertising }
--> Term: { id=3, description=Net due 30 days, due days=30 }
8 | BFI Industries | PO Box 9369
--> Account: { number=521, description=Utilities }
--> Term: { id=3, description=Net due 30 days, due days=30 }
9 | Pacific Gas & Electric | Box 52001
--> Account: { number=521, description=Utilities }
--> Term: { id=3, description=Net due 30 days, due days=30 }
10 | Robbins Mobile Lock And Key | 4669 N Fresno
--> Account: { number=523, description=Building Maintenance }
--> Term: { id=2, description=Net due 20 days, due days=20 }
-----

```

กำหนดให้:

- หากความสัมพันธ์ระหว่าง Vendor และ VendorContract ไม่เป็น null ให้แสดง ชื่อ (name) ของ Vendor ในรูปแบบ “Firstname Lastname” โดยนำข้อมูลจาก VendorContract
- หากความสัมพันธ์ระหว่าง Vendor และ GeneralLedgerAccount ไม่เป็น null ให้แสดงข้อมูลของ GeneralLedgerAccount
- หากความสัมพันธ์ระหว่าง Vendor และ Term ไม่เป็น null ให้แสดงข้อมูลของ Term

กำหนดให้คลาส HibernateConnection.java ดังแสดงต่อไปนี้

```
package /* your package */;

import java.util.Properties;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;

public class HibernateConnection {
    static String url = "jdbc:mysql://localhost:3306/ap";
    static String uname = "root";
    static String pwd = "1234";

    public static SessionFactory sessionFactory;

    public static SessionFactory getSessionFactory() {
        Properties database = new Properties();

        database.setProperty("hibernate.connection.driver_class",
            "com.mysql.cj.jdbc.Driver");
        database.setProperty("hibernate.connection.username", uname);
        database.setProperty("hibernate.connection.password", pwd);
        database.setProperty("hibernate.connection.url", url);
        database.setProperty("hibernate.dialect",
            "org.hibernate.dialect.MySQL5InnoDBDialect");
        database.setProperty("hibernate.dialect.storage_engine", "innodb");
        database.setProperty("hibernate.hbm2ddl.auto", "update");
        Configuration cfg = new Configuration().setProperties(database)
            .addPackage("bean")
            .addAnnotatedClass(/* your entity class */.class)
            .addAnnotatedClass(/* your entity class */.class);
        StandardServiceRegistryBuilder ssrb = new StandardServiceRegistryBuilder()
            .applySettings(cfg.getProperties());
        sessionFactory = cfg.buildSessionFactory(ssrb.build());

        return sessionFactory;
    }
}
```