آیا زبان برنامه نویسی انتخاب شده برنامه نویسی شیء گرا را پشتیبانی می کند؟ در اینصورت در مورد ساختارهای موجود و روش های پیاده سازی اشیاء در حافظه، چندریختی، وراثت، و غیره توضیح دهید.

زبان SQL از برخی ویژگی‌های شی گرایی پشتیبانی می‌کند، اما به طور کامل یک زبان شی گرا نیست. برای مثال، SQL از مفاهیمی نظیر کلاس، وراثت، چندریختی و پیام‌رسانی شی گرا پشتیبانی نمی‌کند. برخی از پایگاه داده‌های رابطه‌ای مانند PostgreSQL، امکان ایجاد و استفاده از اشیا را در SQL فراهم می‌کنند، اما این اشیا همچنان باید در جداول رابطه‌ای ذخیره شوند. بنابراین، SQL را می‌توان یک زبان پرس و جوی شی گرا–رابطه‌ای دانست، اما نه یک زبان برنامه نویسی شی گرا.

– ساختارهای موجود در SQL شامل جداول، ستون‌ها، ردیف‌ها، اندیس‌ها، کلیدها، محدودیت‌ها، نماها، تریگرها، توابع، رویه‌ها، کورسورها و غیره هستند. هر یک از این ساختارها یک نقش خاص در ذخیره، بازیابی و دستکاری داده‌ها دارند.

– روش‌های پیاده‌سازی اشیا در حافظه در SQL بستگی به نوع پایگاه داده و نسخه SQL دارد. برخی از پایگاه داده‌ها مانند PostgreSQL از یک مدل شی گرا–رابطه‌ای استفاده می‌کنند که امکان ایجاد و استفاده از اشیا را در SQL فراهم می‌کند. اما این اشیا همچنان باید در جداول رابطه‌ای ذخیره شوند. هر جدول شامل یک سری ستون‌ها یا Column و یک سری ردیف‌ها یا Row است که هر ردیف یک رکورد یا Record از داده‌ها را نشان می‌دهد. هر ستون یک نام و یک نوع داده دارد و هر رکورد باید مقداری برای هر ستون داشته باشد.برخی از پایگاه داده‌ها مانند Oracle از یک مدل شی گرا–XML استفاده می‌کنند که امکان ذخیره و بازیابی اشیا به صورت XML را دارد.

برای پیاده سازی اشیاء در حافظه در زبان SQL، می‌توان از چند روش استفاده کرد. یک روش این است که هر شیء را به عنوان یک جدول مجزا در نظر بگیریم و ویژگی‌ها و روش‌های آن را به عنوان ستون‌ها و ردیف‌های جدول تعریف کنیم. برای نمونه، اگر شیء دانشجو را بخواهیم در SQL پیاده سازی کنیم، می‌توانیم یک جدول با نام Student داشته باشیم که شامل ستون‌هایی مانند id، name، major و gpa باشد. هر ردیف از این جدول یک نمونه از شیء دانشجو را نشان می‌دهد. برای ایجاد یک شیء دانشجو جدید، می‌توانیم از دستور INSERT استفاده کنیم. برای مثال:

```
INSERT INTO Student (id, name, major, gpa) VALUES (123, 'Ali', 'Computer Science', 3.5);
```

این دستور یک ردیف جدید به جدول Student اضافه می‌کند که مقادیر ستون‌ها را برای شیء دانشجو مشخص می‌کند. برای دسترسی به ویژگی‌ها و روش‌های یک شیء دانشجو، می‌توانیم از دستور SELECT استفاده کنیم. برای مثال:

```
SELECT name, gpa FROM Student WHERE id = 123;
```

این دستور نام و معدل یک شیء دانشجو را بر اساس شناسه آن برمی‌گرداند. برای تغییر ویژگی‌ها و روش‌های یک شیء دانشجو، می‌توانیم از دستور UPDATE استفاده کنیم. برای مثال:

```
UPDATE Student SET gpa = 3.8 WHERE id = 123;
```

این دستور معدل یک شیء دانشجو را بر اساس شناسه آن تغییر می‌دهد. برای حذف یک شیء دانشجو، می‌توانیم از دستور DELETE استفاده کنیم. برای مثال:

```
DELETE FROM Student WHERE id = 123;
```

این دستور ردیف مربوط به شیء دانشجو را بر اساس شناسه آن از جدول Student حذف می‌کند.

روش دیگری که می‌توان برای پیاده سازی اشیاء در حافظه در زبان SQL استفاده کرد، این است که از مفهوم Object-Relational Mapping (ORM) بهره بگیریم. ORM یک تکنیک است که به ما امکان می‌دهد که اشیاء را به صورت رابطه‌ای در پایگاه داده ذخیره و بازیابی کنیم. ORM از یک لایه انتزاعی برای تبدیل داده‌های رابطه‌ای به داده‌های شیءگرا و برعکس استفاده می‌کند. ORM ما را از نوشتن دستورات SQL برای انجام عملیات‌های پایگاه داده رها می‌کند و به ما اجازه می‌دهد که با اشیاء به صورت شیءگرا کار کنیم. ORM می‌تواند به صورت یک کتابخانه یا یک ابزار مجزا پیاده سازی شود. برای نمونه، SQLAlchemy یک کتابخانه ORM برای زبان Python است که از SQL پشتیبانی می‌کند. با استفاده از SQLAlchemy، می‌توانیم یک کلاس برای شیء دانشجو تعریف کنیم و از آن برای ایجاد، تغییر، بازیابی و حذف اشیاء دانشجو در پایگاه داده استفاده کنیم. برای مثال:

```
from sqlalchemy import Column, Integer, String, Float
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker

#Define a class for Student object
Base = declarative_base()
class Student(Base):
__  tablename__ = 'Student'
  id = Column(Integer, primary_key=True)
  name = Column(String)
  major = Column(String)
  gpa = Column(Float)

  def __repr__(self):
    return    f"<Student(id={self.id},    name={self.name},
major={self.major}, gpa={self.gpa})>"

#Create an engine and a session for the database
engine = create_engine('sqlite:///students.db')
Base.metadata.create_all(engine)
Session = sessionmaker(bind=engine)
session = Session()

#Create a new Student object
ali = Student(id=123, name='Ali', major='Computer Science',
gpa=3.5)
```

```
#Add the object to the session and commit it to the database
session.add(ali)
session.commit()

#Query the object from the database
ali = session.query(Student).filter_by(id=123).first()
print(ali)

#Update the object's attribute and commit it to the database
ali.gpa = 3.8
session.commit()

#Delete the object from the session and the database
session.delete(ali)
session.commit()
```

– چندریختی در SQL به این معنی است که یک تابع یا رویه می‌تواند بر اساس تعداد، نوع یا ترتیب پارامترهای ورودی خود، رفتارهای مختلفی داشته باشد. برای مثال، می‌توان یک تابع با نام add را تعریف کرد که برای دو عدد، جمع آن‌ها را برگرداند و برای دو رشته، الحاق آن‌ها را انجام دهد.

– وراثت در SQL به این معنی است که یک جدول می‌تواند ویژگی‌ها و محدودیت‌های یک جدول دیگر را به ارث ببرد. برای مثال، می‌توان یک جدول با نام person را تعریف کرد که شامل ستون‌های name و age باشد و سپس یک جدول با نام student را ایجاد کرد که از جدول person به ارث برده شده باشد و یک ستون دیگر به نام major داشته باشد.

برخی از پایگاه داده‌ها از مفهوم User-Defined Type (UDT) یا نوع تعریف شده توسط کاربر پشتیبانی می‌کنند که به ما امکان می‌دهد که یک نوع داده جدید را با ویژگی‌ها و روش‌های خود تعریف کنیم. این نوع داده می‌تواند از نوع داده‌های موجود در SQL مشتق شود و از ویژگی وراثت بهره ببرد. همچنین، همانطور که گفته شد برخی از پایگاه داده‌ها از مفهوم Object-Relational Mapping (ORM) بهره می‌گیرند که به ما امکان می‌دهد که اشیاء را به صورت رابطه‌ای در پایگاه داده ذخیره و بازیابی کنیم. ORM از یک لایه انتزاعی برای تبدیل داده‌های رابطه‌ای به داده‌های شیءگرا و برعکس استفاده می‌کند. ORM ما را از نوشتن دستورات SQL برای انجام عملیات‌های پایگاه داده رها می‌کند و به ما اجازه می‌دهد که با اشیاء به صورت شیءگرا کار کنیم. ORM می‌تواند به صورت یک کتابخانه یا یک ابزار مجزا پیاده سازی شود. برای نمونه، SQLAlchemy یک کتابخانه ORM برای زبان Python است که از SQL پشتیبانی می‌کند.

منابع:

- https://sariasan.com
- https://www.mongard.ir
- https://www.mobinhost.com
- https://pouyanit.com