

```
from sqlalchemy import Column, Integer, String, ForeignKey, Text, DateTime, Float, Boolean
```

```
from sqlalchemy.orm import relationship, backref
```

```
from datetime import datetime
```

```
from connections import Base
```

```
class Admin(Base):
```

```
    __tablename__ = 'admins'
```

```
    id = Column(Integer, primary_key=True, index=True)
```

```
    username = Column(String(100), unique=True, index=True)
```

```
    password = Column(String(100))
```

```
    def __init__(self, username, password):
```

```
        self.username = username
```

```
        self.password = password
```


```
class User(Base):
```

```
    __tablename__ = 'users'
```

```
    id = Column(Integer, primary_key=True, index=True)
```

```
    username = Column(String(100), unique=True, index=True)
```

```
    password = Column(String(100))
```

```
#  one-to-one StudentProfile
```

```
    profile = relationship("StudentProfile", back_populates="user", uselist=False)
```

```
def __init__(self, username, password):
```

```
    self.username = username
```

```
    self.password = password
```

```
class Course(Base):
```

```
    __tablename__ = 'courses'
```

```
    id = Column(Integer, primary_key=True, index=True)
```

```
    name = Column(String(100), unique=True, index=True)
```

```
    level = Column(String(255))
```

```
    subjects = relationship('Subject', back_populates='course', cascade='all, delete-orphan')
```

```
def __init__(self, name, level):
```

```
    self.name = name
```

```
    self.level = level
```

```
class Subject(Base):
```

```
    __tablename__ = 'subjects'
```

```
    id = Column(Integer, primary_key=True, index=True)
```

```
    name = Column(String(100), index=True)
```

```
    course_id = Column(Integer, ForeignKey('courses.id'))
```

```
course = relationship('Course', back_populates='subjects')
```

```
def __init__(self, name, course_id):
```

```
    self.name = name
```

```
    self.course_id = course_id
```

```
class Quiz(Base):
```

```
    __tablename__ = 'quizzes'
```

```
    id = Column(Integer, primary_key=True)
```

```
    title = Column(String(100), nullable=False)
```

```
    course_id = Column(Integer, ForeignKey('courses.id'))
```

```
    subject_id = Column(Integer, ForeignKey('subjects.id'))
```

```
    duration = Column(Integer, default=30)
```

```
    upload_time = Column(DateTime, default=datetime.utcnow)
```

```
    status = Column(String(50), default='inactive')
```

```
    course = relationship('Course', backref='quizzes')
```

```
    subject = relationship('Subject', backref='quizzes')
```

```
    questions = relationship('Question', cascade='all, delete-orphan', backref='quiz')
```

```
    results = relationship('Result', backref='quiz')
```

```
def __init__(self, title, course_id, subject_id, duration=30, status='inactive'):
```

```
    self.title = title
```

```
    self.course_id = course_id
```

```
self.subject_id = subject_id
```

```
self.duration = duration
```

```
self.status = status
```

```
class Question(Base):
```

```
    __tablename__ = 'questions'
```

```
    id = Column(Integer, primary_key=True)
```

```
    quiz_id = Column(Integer, ForeignKey('quizzes.id'))
```

```
    question_text = Column(Text)
```

```
    option_a = Column(Text)
```

```
    option_b = Column(Text)
```

```
    option_c = Column(Text)
```

```
    option_d = Column(Text)
```

```
    correct_option = Column(String(1))
```

```
    marks = Column(Integer, default=1)
```

```
    image = Column(Text)
```

```
    extra_content = Column(Text)
```

```
    def __init__(self, quiz_id, question_text, option_a, option_b, option_c, option_d,  
correct_option, marks=1, image=None, extra_content=None):
```

```
        self.quiz_id = quiz_id
```

```
        self.question_text = question_text
```

```
        self.option_a = option_a
```

```
        self.option_b = option_b
```

```
self.option_c = option_c
self.option_d = option_d
self.correct_option = correct_option
self.marks = marks
self.image = image
self.extra_content = extra_content
```

```
class Result(Base):
```

```
    __tablename__ = 'results'
```

```
    id = Column(Integer, primary_key=True)
```

```
    student_id = Column(Integer, ForeignKey('student_profiles.id')) #  link to StudentProfile
```


```
    quiz_id = Column(Integer, ForeignKey('quizzes.id'))
```

```
    score = Column(Integer)
```

```
    total_marks = Column(Integer)
```

```
    percentage = Column(Float)
```

```
    taken_on = Column(DateTime, default=datetime.utcnow)
```

```
    student = relationship('StudentProfile', backref='results') #  connect to profile
```

```
    def __init__(self, student_id, quiz_id, score, total_marks, percentage):
```

```
        self.student_id = student_id
```

```
        self.quiz_id = quiz_id
```

```
        self.score = score
```

```
        self.total_marks = total_marks
```

```
self.percentage = percentage
```

```
class Video(Base):
```

```
    __tablename__ = "videos"
```

```
    id = Column(Integer, primary_key=True)
```

```
    title = Column(String(255), nullable=False)
```

```
    drive_url = Column(String(500), nullable=False)
```

```
    course_id = Column(Integer, ForeignKey('courses.id'), nullable=False)
```

```
    subject_id = Column(Integer, ForeignKey('subjects.id'), nullable=False)
```

```
    upload_time = Column(DateTime, default=datetime.utcnow)
```

```
    course = relationship("Course", backref="videos")
```

```
    subject = relationship("Subject", backref="videos")
```

```
class Document(Base):
```

```
    __tablename__ = "documents"
```

```
    id = Column(Integer, primary_key=True)
```

```
    title = Column(String(255), nullable=False)
```

```
    filename = Column(String(255), nullable=False)
```

```
    course_id = Column(Integer, ForeignKey('courses.id'), nullable=False)
```

```
    subject_id = Column(Integer, ForeignKey('subjects.id'), nullable=False)
```

```
    is_active = Column(Boolean, default=True)
```

```
    course = relationship('Course', backref=backref('documents', lazy=True))
```

```
subject = relationship('Subject', backref=backref('documents', lazy=True))
```

```
class StudentProfile(Base):
```

```
    __tablename__ = 'student_profiles'
```

```
    id = Column(Integer, primary_key=True, index=True)
```

```
    full_name = Column(String(255), nullable=False)
```

```
    exam_type = Column(String(50), nullable=False)
```

```
    course_id = Column(Integer, ForeignKey('courses.id'), nullable=False)
```

```
    admission_number = Column(String(50), unique=True, nullable=False)
```

```
    blocked = Column(Boolean, default=False)
```

```
    phone_number = Column(String(50), nullable=False)
```

```
    user_id = Column(Integer, ForeignKey('users.id'), nullable=False)
```

```
    course = relationship("Course", backref="students")
```

```
    #  back reference to User
```

```
    user = relationship("User", back_populates="profile")
```

```
    def __init__(self, full_name, exam_type, course_id, admission_number,  
phone_number, user_id, blocked=False):
```

```
        self.full_name = full_name
```

```
        self.exam_type = exam_type
```

```
        self.course_id = course_id
```

```
self.admission_number = admission_number
```

```
self.phone_number = phone_number
```

```
self.user_id = user_id
```

```
self.blocked = blocked
```

```
class Message(Base):
```

```
    __tablename__ = "messages"
```

```
    id = Column(Integer, primary_key=True)
```

```
    content = Column(Text, nullable=False)
```

```
    created_at = Column(DateTime, default=datetime.utcnow)
```

```
    target_type = Column(String(50), default="all") # "all", "course", "subject"
```

```
    course_id = Column(Integer, ForeignKey("courses.id"), nullable=True)
```

```
    subject_id = Column(Integer, ForeignKey("subjects.id"), nullable=True)
```

```
    course = relationship("Course", backref="messages")
```

```
    subject = relationship("Subject", backref="messages")
```

```
class ActivityLog(Base):
```

```
    __tablename__ = "activity_logs"
```

```
    id = Column(Integer, primary_key=True)
```

```
    student_id = Column(Integer, ForeignKey("student_profiles.id"))
```

```
    activity_type = Column(String(50)) # "video", "document", "exam"
```

```
    is_active = Column(Boolean, default=True) # currently active
```

```
    started_at = Column(DateTime, default=datetime.utcnow)
```



```
student = relationship("StudentProfile", backref="activities")
```