Hive DDL 실습 워크북 — 연습문제 해답 (한국어판)

《Hive 데이터웨어하우스 응용》 제 4 장 정답 모음 (교사 참고용)

> ⚠️ **교사 전용 : ** 이 파일은 연습문제 표준 정답입니다. 수업 전 학생에게 미리 배포하지 마세요.

실습 4-1 연습문제 해답

연습 1 : `practice_db` 데이터베이스 생성

```sql
CREATE DATABASE IF NOT EXISTS practice_db
COMMENT '연습용 데이터베이스'
LOCATION '/user/hive/warehouse/practice_db'
WITH DBPROPERTIES (
  'author'  = '학생이름',
  'subject' = '빅데이터기술'
);
```

**검증 명령어 : **

```sql
DESC DATABASE EXTENDED practice_db;
```

**예상 출력 포함 항목 : **

- `db_name` : practice_db
- `comment` : 연습용 데이터베이스
- `location` : hdfs://호스트명/user/hive/warehouse/practice_db
- `parameters` : {author=학생이름, subject=빅데이터기술}

Hive DDL 实战工作手册 — 练习题解答（中文版）

《Hive 数据仓库应用》第 4 章 答案手册（仅供教师参考）

> ⚠️ **教师专用 : ** 本文件为练习题标准答案，请勿提前发放给学生。

任务 4-1 练习题解答

练习 1 : 创建 `practice_db` 数据库

```sql
CREATE DATABASE IF NOT EXISTS practice_db
COMMENT '练习用数据库'
LOCATION '/user/hive/warehouse/practice_db'
WITH DBPROPERTIES (
  'author'  = '学生姓名',
  'subject' = '大数据技术'
);
```

**验证命令 : **

```sql
DESC DATABASE EXTENDED practice_db;
```

**预期输出包含 : **

- `db_name` : practice_db
- `comment` : 练习用数据库
- `location` : hdfs://主机名/user/hive/warehouse/practice_db
- `parameters` : {author=学生姓名, subject=大数据技术}

연습 2 : "p"로 시작하는 데이터베이스 와일드카드 조회

```sql
SHOW DATABASES LIKE 'p*';
```

**예상 출력 : **

```
practice_db
```

练习 2：模糊查询以 "p" 开头的数据库

```sql
SHOW DATABASES LIKE 'p*';
```

预期输出：

```
practice_db
```

연습 3 : `practice_db` 상세 정보 확인

```sql
DESC DATABASE EXTENDED practice_db;
```

**예상 출력 (주요 항목) : **

```
practice_db  연습용 데이터베이스
hdfs://xxx/user/hive/warehouse/practice_db  ...
Parameters: {author=학생이름, subject=빅데이터기술}
```

练习 3：查看 `practice_db` 详细信息

```sql
DESC DATABASE EXTENDED practice_db;
```

预期输出（关键字段）：

```
practice_db  练习用数据库
hdfs://xxx/user/hive/warehouse/practice_db  ...
Parameters: {author=学生姓名, subject=大数据技术}
```

연습 4 : 새 속성 `semester` 추가

```sql
ALTER DATABASE practice_db
SET DBPROPERTIES ('semester' = '2024-spring');
```

**검증 : **

```sql
DESC DATABASE EXTENDED practice_db;
```

练习 4：添加新属性 `semester`

```sql
ALTER DATABASE practice_db
SET DBPROPERTIES ('semester' = '2024-spring');
```

验证：

```sql
DESC DATABASE EXTENDED practice_db;
```

예상 출력 (속성 부분) :

```
{author=학생이름, subject=빅데이터기술,
semester=2024-spring}
```

연습 5 : `practice_db` 삭제

```sql
DROP DATABASE IF EXISTS practice_db;
```

검증 :

```sql
SHOW DATABASES;
-- practice_db 가 목록에 더 이상 나타나지 않아야 함
```

실습 4-2 연습문제 해답

연습 1 : 내부 테이블 `dept_info` 생성

```sql
CREATE TABLE IF NOT EXISTS unit4db.dept_info (
  dept_id   INT    COMMENT '부서 번호',
  dept_name STRING COMMENT '부서명',
  manager   STRING COMMENT '부서장',
  location  STRING COMMENT '사무실 위치'
)
COMMENT '부서 정보 내부 테이블'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

검증 :

```sql
```

预期输出（属性部分）：

```
{author=学生姓名, subject=大数据技术,
semester=2024-spring}
```

练习 5：删除 `practice_db`

```sql
DROP DATABASE IF EXISTS practice_db;
```

验证：

```sql
SHOW DATABASES;
-- practice_db 不应再出现在列表中
```

任务 4-2 练习题解答

练习 1：创建内部表 `dept_info`

```sql
CREATE TABLE IF NOT EXISTS unit4db.dept_info (
  dept_id   INT    COMMENT '部门编号',
  dept_name STRING COMMENT '部门名称',
  manager   STRING COMMENT '部门经理',
  location  STRING COMMENT '办公地点'
)
COMMENT '部门信息内部表'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

验证：

```sql
```

```
DESC dept_info;
SHOW CREATE TABLE dept_info;
```

**핵심 포인트 : ** 필드 구분자가 탭(`\t`)임에 주의. 쉼표 구분자와의 차이를 정확히 이해해야 합니다.

연습 2 : 외부 테이블 `salary_grade_ext` 생성

```sql
CREATE EXTERNAL TABLE IF NOT EXISTS
unit4db.salary_grade_ext (
  grade_id    INT     COMMENT '급여 등급',
  min_salary DOUBLE COMMENT '최저 급여',
  max_salary DOUBLE COMMENT '최고 급여',
  grade_name STRING COMMENT '등급명'
)
COMMENT '급여 등급 외부 테이블'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION
'/user/hive/warehouse/unit4db/salary_grade';
```

**검증 : **

```sql
DESC FORMATTED salary_grade_ext;
-- Table Type 이 EXTERNAL_TABLE 인지 확인
-- Location 이
/user/hive/warehouse/unit4db/salary_grade 인지 확인
```

연습 3 : `dept_info`에 `phone`, `email` 컬럼 추가

```sql
ALTER TABLE unit4db.dept_info
ADD COLUMNS (
```

```
DESC dept_info;
SHOW CREATE TABLE dept_info;
```

**要点 : ** 字段分隔符为制表符 `\t`，注意与逗号分隔的区别。

练习 2 : 创建外部表 `salary_grade_ext`

```sql
CREATE EXTERNAL TABLE IF NOT EXISTS
unit4db.salary_grade_ext (
  grade_id    INT     COMMENT '薪资等级',
  min_salary DOUBLE COMMENT '最低薪资',
  max_salary DOUBLE COMMENT '最高薪资',
  grade_name STRING COMMENT '等级名称'
)
COMMENT '薪资等级外部表'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION
'/user/hive/warehouse/unit4db/salary_grade';
```

**验证 : **

```sql
DESC FORMATTED salary_grade_ext;
-- 确认 Table Type 为 EXTERNAL_TABLE
-- 确认 Location 为
/user/hive/warehouse/unit4db/salary_grade
```

练习 3 : 向 `dept_info` 添加 `phone` 和 `email` 列

```sql
ALTER TABLE unit4db.dept_info
ADD COLUMNS (
```

```
  phone STRING COMMENT '연락처',
  email STRING COMMENT '부서 이메일'
);
```

**검증 : **

```sql
DESC dept_info;
```

**예상 출력 (새 컬럼 포함) : **

```
dept_id   int      부서 번호
dept_name string   부서명
manager   string   부서장
location  string   사무실 위치
phone     string   연락처       <- 새로 추가
email     string   부서 이메일   <- 새로 추가
```

연습 4 : `manager` 컬럼을 `dept_manager`로 이름 변경 및 위치 이동

```sql
ALTER TABLE unit4db.dept_info
CHANGE manager dept_manager STRING
COMMENT '부서장'
AFTER dept_name;
```

**검증 : **

```sql
DESC dept_info;
```

**예상 컬럼 순서 : **

```
dept_id → dept_name → dept_manager → location →
```

```
  phone STRING COMMENT '联系电话',
  email STRING COMMENT '部门邮箱'
);
```

**验证 : **

```sql
DESC dept_info;
```

**预期输出 (包含新增列) : **

```
dept_id   int      部门编号
dept_name string   部门名称
manager   string   部门经理
location  string   办公地点
phone     string   联系电话     <- 新增
email     string   部门邮箱     <- 新增
```

练习 4 : 将 `manager` 列重命名为 `dept_manager` 并移动位置

```sql
ALTER TABLE unit4db.dept_info
CHANGE manager dept_manager STRING
COMMENT '部门经理'
AFTER dept_name;
```

**验证 : **

```sql
DESC dept_info;
```

**预期列顺序 : **

```
dept_id -> dept_name -> dept_manager -> location ->
```

phone → email
```

---

### 연습 5：내부 테이블과 외부 테이블의 Table Type 비교

**내부 테이블 조회：**

```sql
DESC FORMATTED unit4db.dept_info;
```

외부 테이블 조회：

```sql
DESC FORMATTED unit4db.salary_grade_ext;
```

핵심 차이점：

항목	내부 테이블 (dept_info)	외부 테이블 (salary_grade_ext)
`Table Type`	`MANAGED_TABLE`	`EXTERNAL_TABLE`
테이블 삭제 시 데이터 처리	메타데이터 + HDFS 데이터 모두 삭제	메타데이터만 삭제, HDFS 데이터 보존

교사 설명 포인트：

- 외부 테이블은 다중 시스템 데이터 공유 시나리오에 적합 (예：Spark 와 Hive 가 동일한 데이터셋을 공동으로 읽는 경우)
- 내부 테이블은 Hive 가 단독으로 데이터를 관리하는 시나리오에 적합
- 실제 운영 환경에서는 데이터 실수 삭제를 방지하기 위해 대부분의 업무 테이블에 외부 테이블 권장

phone -> email
```

练习 5：对比内外部表的 Table Type

内部表查询：

```sql
DESC FORMATTED unit4db.dept_info;
```

外部表查询：

```sql
DESC FORMATTED unit4db.salary_grade_ext;
```

关键区别：

字段	内部表 (dept_info)	外部表 (salary_grade_ext)
`Table Type`	`MANAGED_TABLE`	`EXTERNAL_TABLE`
删除表时数据处理	元数据 + HDFS 数据均删除	只删除元数据，HDFS 数据保留

教师说明要点：

- 外部表适合多系统共享数据的场景（如 Spark 和 Hive 共同读取同一数据集）
- 内部表适合 Hive 独自管理数据的场景
- 实际生产中，大多数业务表建议使用外部表，以防误删数据

실습 4-3 연습문제 해답

연습 1 : 3 단계 파티션 테이블 `attendance` 생성

```sql
CREATE TABLE IF NOT EXISTS unit4db.attendance (
  att_id     INT    COMMENT '기록 ID',
  emp_id     INT    COMMENT '직원 ID',
  att_status STRING COMMENT '출근 상태'
)
COMMENT '직원 출근 기록 파티션 테이블'
PARTITIONED BY (
  att_year  STRING COMMENT '연도',
  att_month STRING COMMENT '월',
  att_day   STRING COMMENT '일'
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

검증 :

```sql
DESC FORMATTED attendance;
-- Partition Information 섹션에서 세 개의 파티션 필드 확인
```

연습 2 : 파티션 추가

```sql
ALTER TABLE unit4db.attendance
ADD IF NOT EXISTS
PARTITION (att_year='2024', att_month='01', att_day='15')
PARTITION (att_year='2024', att_month='01', att_day='16')
PARTITION (att_year='2024', att_month='02', att_day='01');
```

任务 4-3 练习题解答

练习 1 : 创建三级分区表 `attendance`

```sql
CREATE TABLE IF NOT EXISTS unit4db.attendance (
  att_id     INT    COMMENT '记录 ID',
  emp_id     INT    COMMENT '员工 ID',
  att_status STRING COMMENT '出勤状态'
)
COMMENT '员工出勤记录分区表'
PARTITIONED BY (
  att_year  STRING COMMENT '年份',
  att_month STRING COMMENT '月份',
  att_day   STRING COMMENT '日期'
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

验证 :

```sql
DESC FORMATTED attendance;
-- 查看 Partition Information 部分，确认三个分区字段
```

练习 2 : 添加分区

```sql
ALTER TABLE unit4db.attendance
ADD IF NOT EXISTS
PARTITION (att_year='2024', att_month='01', att_day='15')
PARTITION (att_year='2024', att_month='01', att_day='16')
PARTITION (att_year='2024', att_month='02', att_day='01');
```

**검증 : **

```sql
SHOW PARTITIONS unit4db.attendance;
```

**예상 출력 : **

```
att_year=2024/att_month=01/att_day=15
att_year=2024/att_month=01/att_day=16
att_year=2024/att_month=02/att_day=01
```

연습 3 : 특정 조건의 파티션 조회

```sql
SHOW PARTITIONS attendance
PARTITION(att_year='2024', att_month='01');
```

**예상 출력 : **

```
att_year=2024/att_month=01/att_day=15
att_year=2024/att_month=01/att_day=16
```

연습 4 : 파티션 삭제 및 검증

```sql
ALTER TABLE unit4db.attendance
DROP IF EXISTS PARTITION (att_year='2024',
att_month='02');
```

**검증 : **

```sql
SHOW PARTITIONS unit4db.attendance;
```

**예상 출력 (2 월 파티션 삭제됨) : **

**验证 : **

```sql
SHOW PARTITIONS unit4db.attendance;
```

**预期输出 : **

```
att_year=2024/att_month=01/att_day=15
att_year=2024/att_month=01/att_day=16
att_year=2024/att_month=02/att_day=01
```

练习 3 : 查询指定条件的分区

```sql
SHOW PARTITIONS attendance
PARTITION(att_year='2024', att_month='01');
```

**预期输出 : **

```
att_year=2024/att_month=01/att_day=15
att_year=2024/att_month=01/att_day=16
```

练习 4 : 删除分区并验证

```sql
ALTER TABLE unit4db.attendance
DROP IF EXISTS PARTITION (att_year='2024',
att_month='02');
```

**验证 : **

```sql
SHOW PARTITIONS unit4db.attendance;
```

**预期输出 (2 月分区已删除) : **

```
att_year=2024/att_month=01/att_day=15
att_year=2024/att_month=01/att_day=16
```

실습 4-4 연습문제 해답

연습 1 : 직원 급여 버킷 테이블 생성

```sql
CREATE TABLE IF NOT EXISTS unit4db.emp_salary_bucketed (
  emp_id   INT    COMMENT '직원 ID',
  emp_name STRING COMMENT '직원 이름',
  dept     STRING COMMENT '소속 부서',
  salary   DOUBLE COMMENT '급여'
)
COMMENT '직원 급여 버킷 테이블 (부서별 3 버킷) '
CLUSTERED BY (dept) SORTED BY (salary ASC) INTO 3 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

연습 2 : 임시 테이블 생성 및 데이터 로드 후 버킷 테이블에 삽입

1 단계 : 임시 테이블 생성 및 데이터 로드

```sql
CREATE TABLE IF NOT EXISTS unit4db.emp_salary_temp
(
  emp_id    INT,
  emp_name  STRING,
  dept      STRING,
  salary    DOUBLE,
  hire_date STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

LOAD DATA LOCAL INPATH
```

```
att_year=2024/att_month=01/att_day=15
att_year=2024/att_month=01/att_day=16
```

任务 4-4 练习题解答

练习 1 : 创建员工薪资分桶表

```sql
CREATE TABLE IF NOT EXISTS unit4db.emp_salary_bucketed (
  emp_id   INT    COMMENT '员工 ID',
  emp_name STRING COMMENT '员工姓名',
  dept     STRING COMMENT '所属部门',
  salary   DOUBLE COMMENT '薪资'
)
COMMENT '员工薪资分桶表 (按部门分 3 桶) '
CLUSTERED BY (dept) SORTED BY (salary ASC) INTO 3 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

练习 2 : 创建临时表并加载数据，再写入分桶表

步骤 1 : 创建临时表并加载数据

```sql
CREATE TABLE IF NOT EXISTS unit4db.emp_salary_temp
(
  emp_id    INT,
  emp_name  STRING,
  dept      STRING,
  salary    DOUBLE,
  hire_date STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

LOAD DATA LOCAL INPATH
```

```
'/usr/datas/unit4/employees.txt'
INTO TABLE unit4db.emp_salary_temp;
```

2 단계 : INSERT INTO ... SELECT 로 버킷 테이블에 삽입

```sql
SET hive.enforce.bucketing = true;

INSERT INTO TABLE unit4db.emp_salary_bucketed
SELECT emp_id, emp_name, dept, salary
FROM unit4db.emp_salary_temp;
```

연습 3 : 버킷 테이블 메타데이터 검증

```sql
DESC FORMATTED unit4db.emp_salary_bucketed;
```

**확인해야 할 주요 항목 : **

```
Num Buckets:    3
Bucket Columns:  [dept]
Sort Columns:    [Order: ASC]
```

연습 4 : 부서별 직원 수 통계

```sql
SELECT dept, COUNT(*) AS emp_count
FROM unit4db.emp_salary_bucketed
GROUP BY dept;
```

**예상 출력 (샘플 데이터 기준) : **

```
技术部    4
市场部    3
```

```
'/usr/datas/unit4/employees.txt'
INTO TABLE unit4db.emp_salary_temp;
```

步骤 2 : 通过 INSERT INTO ... SELECT 写入分桶表

```sql
SET hive.enforce.bucketing = true;

INSERT INTO TABLE unit4db.emp_salary_bucketed
SELECT emp_id, emp_name, dept, salary
FROM unit4db.emp_salary_temp;
```

练习 3 : 验证分桶表元数据

```sql
DESC FORMATTED unit4db.emp_salary_bucketed;
```

**需确认的关键字段 : **

```
Num Buckets:    3
Bucket Columns:  [dept]
Sort Columns:    [Order: ASC]
```

练习 4 : 按部门统计员工数量

```sql
SELECT dept, COUNT(*) AS emp_count
FROM unit4db.emp_salary_bucketed
GROUP BY dept;
```

**预期输出 (基于示例数据) : **

```
技术部    4
市场部    3
```

人事部 2
财务部 1
```

**교사 설명 : ** 해시 버킷팅으로 인해 각 부서의 직원이 같은 버킷 또는 다른 버킷에 배분될 수 있습니다. 버킷 필드 `dept`는 어떤 행이 같은 버킷에 들어갈지를 결정하지만, 쿼리 결과의 정확성에는 영향을 미치지 않습니다.

---

## 실습 4-5 연습문제 해답

### 연습 1 : 뷰 `v_dept_emp_list` 생성

```sql
CREATE VIEW IF NOT EXISTS unit4db.v_dept_emp_list
COMMENT '부서별 직원 목록 뷰'
AS
SELECT
  dept        AS department,
  emp_id      AS employee_id,
  emp_name    AS employee_name,
  salary
FROM unit4db.employees
ORDER BY dept ASC, salary DESC;
```

**검증 : **

```sql
SELECT * FROM unit4db.v_dept_emp_list;
```

연습 2 : 성(省份)별 주문 통계 뷰 생성

```sql
CREATE VIEW IF NOT EXISTS
unit4db.v_province_orders
COMMENT '성별 주문 통계 뷰'
AS
```

人事部 2
财务部 1
```

教师说明： 由于哈希分桶，各部门的员工可能被分配到同一个桶或不同的桶中，分桶字段 `dept` 决定哪些行进入同一个桶，但不影响查询结果的正确性。

任务 4-5 练习题解答

练习 1：创建视图 `v_dept_emp_list`

```sql
CREATE VIEW IF NOT EXISTS unit4db.v_dept_emp_list
COMMENT '各部门员工列表视图'
AS
SELECT
  dept        AS department,
  emp_id      AS employee_id,
  emp_name    AS employee_name,
  salary
FROM unit4db.employees
ORDER BY dept ASC, salary DESC;
```

验证：

```sql
SELECT * FROM unit4db.v_dept_emp_list;
```

练习 2：创建省份订单统计视图

```sql
CREATE VIEW IF NOT EXISTS
unit4db.v_province_orders
COMMENT '各省份订单统计视图'
AS
```

```
SELECT
  province,
  COUNT(order_id) AS order_count,
  SUM(amount)     AS total_amount
FROM unit4db.orders_bucketed
GROUP BY province;
```

검증 :

```sql
SELECT * FROM unit4db.v_province_orders
ORDER BY total_amount DESC;
```

연습 3 : 뷰 `v_dept_emp_list` 수정 (부서 평균 급여보다 높은 직원만 표시)

```sql
ALTER VIEW unit4db.v_dept_emp_list
AS
SELECT
  e.dept        AS department,
  e.emp_id      AS employee_id,
  e.emp_name    AS employee_name,
  e.salary
FROM unit4db.employees e
INNER JOIN (
  SELECT dept, AVG(salary) AS avg_sal
  FROM unit4db.employees
  GROUP BY dept
) dept_avg ON e.dept = dept_avg.dept
WHERE e.salary > dept_avg.avg_sal
ORDER BY e.dept ASC, e.salary DESC;
```

검증 :

```sql
SELECT * FROM unit4db.v_dept_emp_list;
```

연습 4 : DESC FORMATTED 으로 뷰 정의 확인

```
SELECT
  province,
  COUNT(order_id) AS order_count,
  SUM(amount)     AS total_amount
FROM unit4db.orders_bucketed
GROUP BY province;
```

验证 :

```sql
SELECT * FROM unit4db.v_province_orders
ORDER BY total_amount DESC;
```

练习 3 : 修改视图 `v_dept_emp_list` (仅显示高于部门平均薪资的员工)

```sql
ALTER VIEW unit4db.v_dept_emp_list
AS
SELECT
  e.dept        AS department,
  e.emp_id      AS employee_id,
  e.emp_name    AS employee_name,
  e.salary
FROM unit4db.employees e
INNER JOIN (
  SELECT dept, AVG(salary) AS avg_sal
  FROM unit4db.employees
  GROUP BY dept
) dept_avg ON e.dept = dept_avg.dept
WHERE e.salary > dept_avg.avg_sal
ORDER BY e.dept ASC, e.salary DESC;
```

验证 :

```sql
SELECT * FROM unit4db.v_dept_emp_list;
```

练习 4 : 使用 DESC FORMATTED 查看视图定义

```sql
DESC FORMATTED unit4db.v_dept_emp_list;
```

**확인해야 할 주요 항목 : **

- `Table Type` : `VIRTUAL_VIEW`
- `View Original Text` : 뷰의 원본 쿼리 구문 표시
- `View Expanded Text` : 완전히 전개된 쿼리 구문 표시

종합 연습문제 해답

요구 1 : 데이터베이스 작업

```sql
-- 데이터베이스 생성
CREATE DATABASE IF NOT EXISTS hr_db
COMMENT '인사 데이터베이스'
LOCATION '/user/hive/warehouse/hr_db'
WITH DBPROPERTIES (
  'dept'    = 'HR',
  'version' = '2.0'
);

-- 속성 수정: last_update 추가
ALTER DATABASE hr_db
SET DBPROPERTIES ('last_update' = '2024-06-01');

-- 검증
DESC DATABASE EXTENDED hr_db;
```

요구 2 : 테이블 작업

```sql
-- 내부 테이블 hr_employees 생성
CREATE TABLE IF NOT EXISTS hr_db.hr_employees (
  emp_id   INT    COMMENT '직원 ID',
  emp_name STRING COMMENT '직원 이름',
  age      INT    COMMENT '나이',
  dept_id  INT    COMMENT '부서 ID',
```

```sql
DESC FORMATTED unit4db.v_dept_emp_list;
```

需确认的关键字段：

- `Table Type` : `VIRTUAL_VIEW`
- `View Original Text` : 显示视图的原始查询语句
- `View Expanded Text` : 显示展开后的完整查询语句

综合练习题解答

要求 1 : 数据库操作

```sql
-- 创建数据库
CREATE DATABASE IF NOT EXISTS hr_db
COMMENT '人力资源数据库'
LOCATION '/user/hive/warehouse/hr_db'
WITH DBPROPERTIES (
  'dept'    = 'HR',
  'version' = '2.0'
);

-- 修改属性，添加 last_update
ALTER DATABASE hr_db
SET DBPROPERTIES ('last_update' = '2024-06-01');

-- 验证
DESC DATABASE EXTENDED hr_db;
```

要求 2 : 表操作

```sql
-- 创建内部表 hr_employees
CREATE TABLE IF NOT EXISTS hr_db.hr_employees (
  emp_id   INT    COMMENT '员工 ID',
  emp_name STRING COMMENT '员工姓名',
  age      INT    COMMENT '年龄',
  dept_id  INT    COMMENT '部门 ID',
```

```
  salary    DOUBLE COMMENT '급여',
  city      STRING COMMENT '소재 도시'
)
COMMENT '인사 직원 테이블 (내부 테이블) '
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- 외부 테이블 hr_departments 생성
CREATE EXTERNAL TABLE IF NOT EXISTS
hr_db.hr_departments (
  dept_id    INT    COMMENT '부서 ID',
  dept_name STRING COMMENT '부서명',
  dept_head STRING COMMENT '부서장'
)
COMMENT '부서 정보 외부 테이블'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/user/hive/warehouse/hr_db/departments';
```

요구 3 : 파티션 테이블

```sql
-- 파티션 테이블 생성
CREATE TABLE IF NOT EXISTS hr_db.hr_salary_history
(
  emp_id      INT    COMMENT '직원 ID',
  base_salary DOUBLE COMMENT '기본 급여',
  bonus       DOUBLE COMMENT '보너스'
)
COMMENT '직원 급여 이력 파티션 테이블'
PARTITIONED BY (
  year    STRING COMMENT '연도',
  quarter STRING COMMENT '분기'
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- 파티션 추가
ALTER TABLE hr_db.hr_salary_history
ADD IF NOT EXISTS
PARTITION (year='2024', quarter='Q1')
PARTITION (year='2024', quarter='Q2');
```

```
  salary    DOUBLE COMMENT '薪资',
  city      STRING COMMENT '所在城市'
)
COMMENT '人力资源员工表（内部表）'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- 创建外部表 hr_departments
CREATE EXTERNAL TABLE IF NOT EXISTS
hr_db.hr_departments (
  dept_id    INT    COMMENT '部门 ID',
  dept_name STRING COMMENT '部门名称',
  dept_head STRING COMMENT '部门负责人'
)
COMMENT '部门信息外部表'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/user/hive/warehouse/hr_db/departments';
```

要求 3：分区表

```sql
-- 创建分区表
CREATE TABLE IF NOT EXISTS hr_db.hr_salary_history
(
  emp_id      INT    COMMENT '员工 ID',
  base_salary DOUBLE COMMENT '基本薪资',
  bonus       DOUBLE COMMENT '奖金'
)
COMMENT '员工薪资历史分区表'
PARTITIONED BY (
  year    STRING COMMENT '年份',
  quarter STRING COMMENT '季度'
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- 添加分区
ALTER TABLE hr_db.hr_salary_history
ADD IF NOT EXISTS
PARTITION (year='2024', quarter='Q1')
PARTITION (year='2024', quarter='Q2');
```

```sql
-- 검증
SHOW PARTITIONS hr_db.hr_salary_history;
```

요구 4 : 버킷 테이블

```sql
CREATE TABLE IF NOT EXISTS
hr_db.hr_employees_bucketed (
  emp_id   INT    COMMENT '직원 ID',
  emp_name STRING COMMENT '직원 이름',
  age      INT    COMMENT '나이',
  dept_id  INT    COMMENT '부서 ID',
  salary   DOUBLE COMMENT '급여',
  city     STRING COMMENT '소재 도시'
)
COMMENT '직원 버킷 테이블 (dept_id 별 4 버킷) '
CLUSTERED BY (dept_id) SORTED BY (salary DESC)
INTO 4 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

요구 5 : 뷰

```sql
-- 뷰 1 : 고급 직원 (급여 12000 초과)
CREATE VIEW IF NOT EXISTS hr_db.v_senior_employees
COMMENT '급여 12000 초과 직원 뷰'
AS
SELECT * FROM hr_db.hr_employees
WHERE salary > 12000;

-- 뷰 2 : 도시별 통계
CREATE VIEW IF NOT EXISTS hr_db.v_city_stats
COMMENT '도시별 직원 통계 뷰'
AS
SELECT
  city,
  COUNT(emp_id) AS emp_count,
  AVG(salary)   AS avg_salary
FROM hr_db.hr_employees
GROUP BY city;
```

```sql
-- 验证
SHOW PARTITIONS hr_db.hr_salary_history;
```

要求 4 : 分桶表

```sql
CREATE TABLE IF NOT EXISTS
hr_db.hr_employees_bucketed (
  emp_id   INT    COMMENT '员工 ID',
  emp_name STRING COMMENT '员工姓名',
  age      INT    COMMENT '年龄',
  dept_id  INT    COMMENT '部门 ID',
  salary   DOUBLE COMMENT '薪资',
  city     STRING COMMENT '所在城市'
)
COMMENT '员工分桶表 (按 dept_id 分 4 桶) '
CLUSTERED BY (dept_id) SORTED BY (salary DESC)
INTO 4 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

要求 5 : 视图

```sql
-- 视图 1 : 高薪员工
CREATE VIEW IF NOT EXISTS hr_db.v_senior_employees
COMMENT '薪资高于 12000 的员工视图'
AS
SELECT * FROM hr_db.hr_employees
WHERE salary > 12000;

-- 视图 2 : 城市统计
CREATE VIEW IF NOT EXISTS hr_db.v_city_stats
COMMENT '各城市员工统计视图'
AS
SELECT
  city,
  COUNT(emp_id) AS emp_count,
  AVG(salary)   AS avg_salary
FROM hr_db.hr_employees
GROUP BY city;
```

요구 6 : 조회 및 검증

```sql
-- 모든 객체 확인
USE hr_db;
SHOW TABLES;
SHOW VIEWS;
SHOW PARTITIONS hr_salary_history;

-- 내부/외부 테이블 타입 비교
DESC FORMATTED hr_employees;      -- Table Type: MANAGED_TABLE
DESC FORMATTED hr_departments;    -- Table Type: EXTERNAL_TABLE
```

교사 채점 기준

채점 배점 제안

실습	배점	핵심 평가 포인트
실습 4-1	10 점	CREATE DATABASE 문법, DBPROPERTIES, DROP CASCADE
실습 4-2	25 점	내외부 테이블 차이, ALTER TABLE 각 조작, LOAD DATA LOCAL
실습 4-3	20 점	PARTITIONED BY 문법, 파티션 필드 중복 금지, ADD/DROP PARTITION
실습 4-4	20 점	CLUSTERED BY/INTO BUCKETS, INSERT 로 버킷 테이블 로드
실습 4-5	15 점	CREATE VIEW, ALTER VIEW 정의 수정, VIRTUAL_VIEW 타입
종합연습	10 점	모든 DDL 조작의 종합 활용

자주 발생하는 오류 안내

要求 6：查询验证

```sql
-- 查看所有对象
USE hr_db;
SHOW TABLES;
SHOW VIEWS;
SHOW PARTITIONS hr_salary_history;

-- 对比内外部表类型
DESC FORMATTED hr_employees;      -- Table Type: MANAGED_TABLE
DESC FORMATTED hr_departments;    -- Table Type: EXTERNAL_TABLE
```

教师批改要点

评分标准建议

任务	分值	关键考察点
任务 4-1	10 分	CREATE DATABASE 语法、DBPROPERTIES、DROP CASCADE
任务 4-2	25 分	内外部表区别、ALTER TABLE 各操作、LOAD DATA LOCAL
任务 4-3	20 分	PARTITIONED BY 语法、分区字段不重名、ADD/DROP PARTITION
任务 4-4	20 分	CLUSTERED BY/INTO BUCKETS、通过 INSERT 加载分桶表
任务 4-5	15 分	CREATE VIEW、ALTER VIEW 修改定义、VIRTUAL_VIEW 类型
综合练习	10 分	综合运用所有 DDL 操作

常见错误提示

1. **파티션 필드와 테이블 컬럼명 중복** → 파티션 테이블 생성 시 오류 발생, 컬럼명 확인 필요
2. **버킷 테이블에 LOAD DATA 직접 사용** → 데이터가 버킷별로 분배되지 않음, INSERT INTO ... SELECT 방식 사용 필요
3. **비어있지 않은 데이터베이스 삭제 시 CASCADE 누락** → 오류 발생, CASCADE 옵션 추가 필요
4. **외부 테이블에 EXTERNAL 키워드 누락** → 실제로 내부 테이블이 생성됨
5. **내부 테이블의 LOCATION 이 기본 경로 밖에 있는 경우** → 문법적으로 허용되지만 데이터 저장 위치를 정확히 파악해야 함

본 정답 모음은 워크북과 함께 사용하며, 교사 참고용입니다.

1. **分区字段与表字段重名** → 创建分区表时报错，需检查列名
2. **直接 LOAD DATA 到分桶表** → 数据不会按桶分配，需通过 INSERT INTO ... SELECT

3. **删除非空数据库未加 CASCADE** → 报错，需加 CASCADE 选项

4. **外部表忘加 EXTERNAL 关键字** → 实际创建的是内部表

5. **内部表 LOCATION 不在默认路径** → 合法但需知道数据存放位置

本答案手册与工作手册配套使用，仅供教师参考。