HD·EDUCATION

HD EDUCATION

FIT2094

TUTOR:Joey

HD·EDUCATION

全球累计服务用户超十万

扫码关注HD·EDU公众号
获取更多留学新资讯

· 让海外学习更轻松 ·

关于
HD EDUCATION

HD · EDUCATION（简称HD·EDU）成立于2018年1月，拥有学业辅导和职业规划两大核心业务。从创办伊始就秉承着 “让年轻人成为知识的生产者、传播者、受惠者” 的使命，坚持从留学生的角度出发，为他们量身制定属于他们的课程。“成为最受年轻人喜爱的教育品牌“一直是我们的不懈追求。

截止2020年，我们的Tutor人数已达1300人，业务范围涵盖了澳大利亚、新西兰、美国、英国4个国家的40多所高等校，为15万留学生提供了优质的学习辅导服务，成为澳大利亚华人留学生覆盖人数最多的在线教育学习平台。

HD·EDU的成长有你陪伴

课后，如果您有任何建议和意见，我们都非常欢迎您联系小助手分享您的想法，给予我们改进和提高的机会！

感谢您参与HD Education的辅导课程！

TUTOR
Self-Introduction
自我介绍

TUTOR: Joey

#

1.🎓 Monash University

2. 📄 Bachelor of Computer Science in Data Science

Master of Information Technology

3. 📖 三年数据库相关教学经验

4. 👩🏻‍💻 从学生角度出发讲解，耐心负责

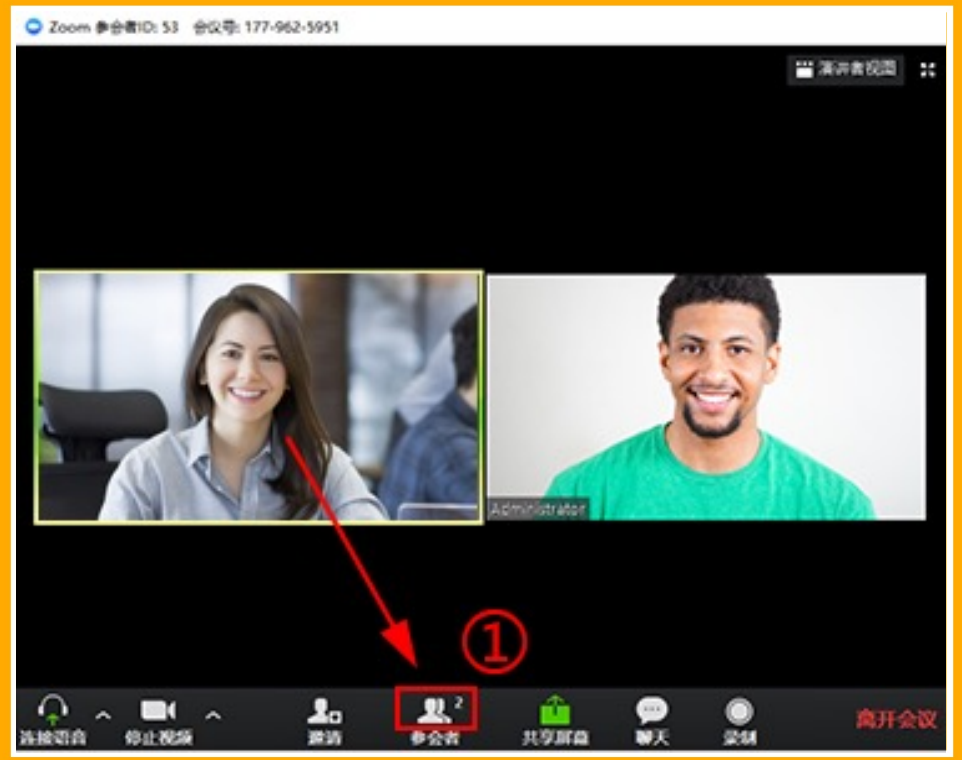
5. ❤️ 吸猫 乐高 旅游

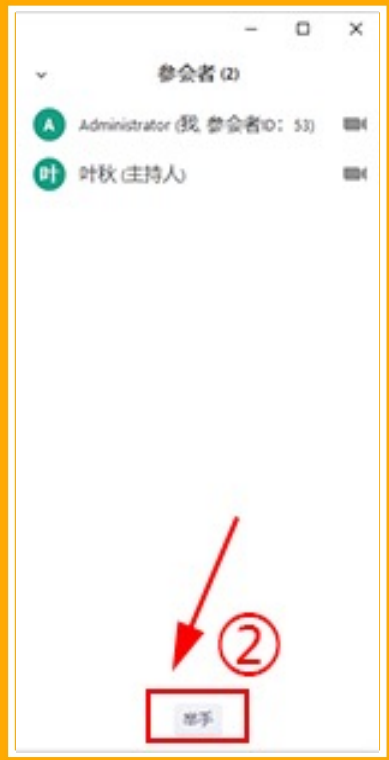
6. 👩🏻‍💻目前在某互联网大厂任数据工程师

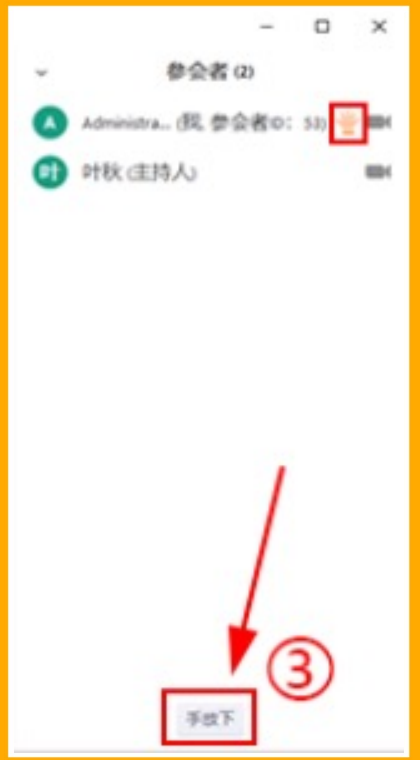
HD·EDUCATION

同学们有问题怎么办?

方法一:举手





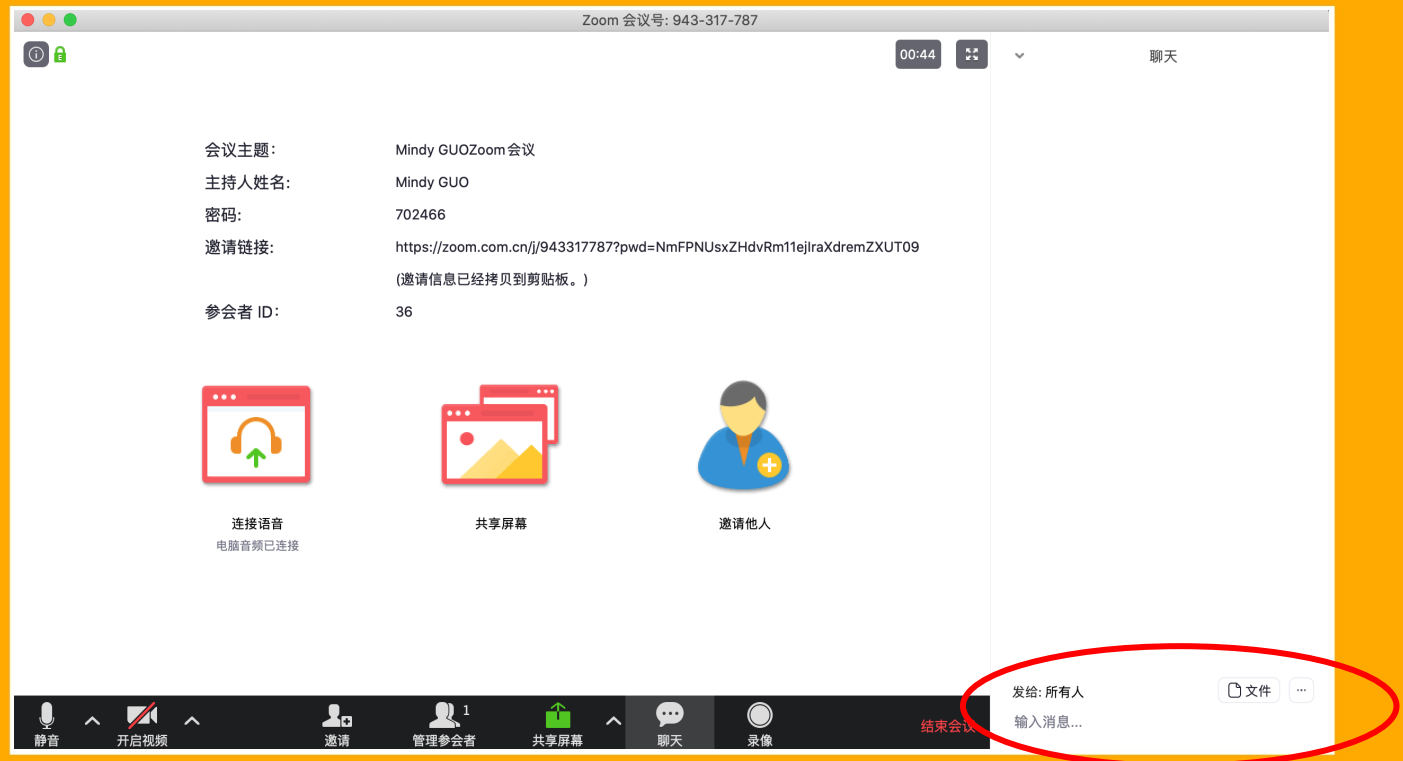


1.点【参会者】
2.点【举手】即可与老师实时互动
3.问题被解答了还可以【手放下】

HD·EDUCATION

同学们有问题怎么办?

方法二:
文字提问



红圈处输入问题提问

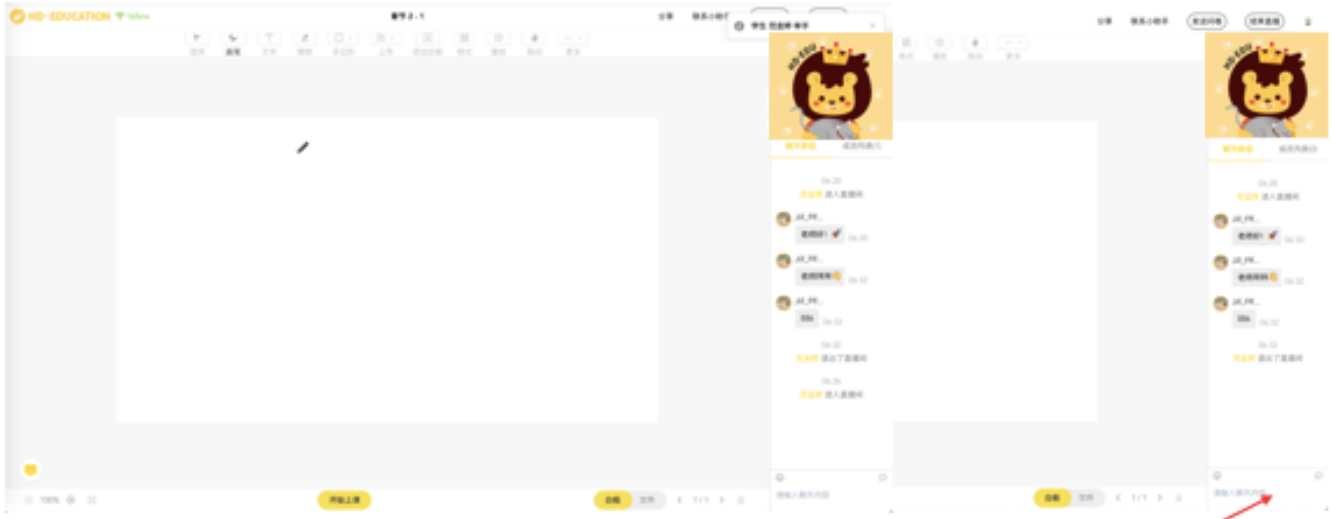
HD·EDUCATION

同学们有问题怎么办?

直播平台

互动方法

直播平台：举手+聊天室提问



点【参会者】再点【举手】，即可与老师实时互动！ 在此输入你想问的问题

问题被解答了还可以【手放下】

学科特点及学习方法

学科特点：

1. 考试题量大，题目长
2. 知识点清晰，相互关联性大
3. 注重细节

学习方法：

1. 养成计时答题的习惯
2. 充分利用学校tutorial资料

备考方法 & Tips：

1. 多读材料，养成能快速在材料定位关键信息的能力
2. 多练代码，保持手感
3. 做题一定要计时！！！

HD·EDUCATION

CONTENT

课程目录

上周回顾

重要程度：⭐⭐⭐
难易程度：⭐⭐⭐⭐

1、Transaction Management

- Atomicity
- 要么全发生，要么全不发生

- Consistency
- 要从一个完整状态转到另一个完整状态

- Isolation
- 多个transaction之间的状态互不干扰

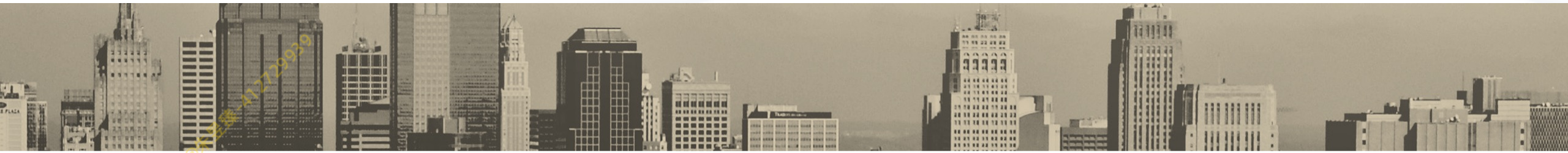
- Durability
- Transaction结束后，结果应一段时间停留，直到下一次transaction

- Shared lock
- 所有transaction都只能读，不能写

- Exclusive lock
- 某一个transaction上此锁后只能它写和读，其他transaction不能进行任何读写操作

本章节知识点

知识点1 SQL QUERY



HD·EDUCATION

知识点讲解

SQL – DML (insert data)

重要程度：⭐⭐⭐⭐
难易程度：⭐⭐

- 要和create table里attribute的顺序一一对应

INSERT INTO *table [(column [, column...])]*
VALUES *(value [, value...]);*

```
INSERT INTO unit VALUES ('FIT9132', 'Databases');
INSERT INTO student VALUES (112233, 'Wild','Wilbur',
                            '01-Jan-1995')
```

Role of: to_date and to_char

SQL – DML (Update)

重要程度：★★★★★
难易程度：★★★★

UPDATE table
SET column1 = xxx, column1 = yyy WHERE condition;

```
UPDATE enrolment
SET mark = 80,
      grade ='HD'
WHERE sno = 112233
  and ……
```

```
UPDATE enrolment
SET mark = 85
WHERE unit_code = (SELECT unit_code FROM unit WHERE
                          unit_name='Introduction to databases')
        AND mark = 80;
```

SQL – DML (Delete)

重要程度：★★★★★
难易程度：★★★★

DELETE FROM table WHERE condition;

```
DELETE FROM enrolment
WHERE sno='112233'
      AND
          unit_code= (SELECT unit_code FROM unit
                      WHERE unit_name='Introduction to Database' )
      AND
          semester='1'
      AND
          year='2012';
```

SQL – DML (COMMIT & ROLLBACK)

重要程度：⭐⭐⭐⭐
难易程度：⭐⭐

- COMMIT: 保存
- ROLLBACK: 撤回

```
INSERT INTO enrolment VALUES (112233,
                  'FIT9132',1,2018,45,'N');
INSERT INTO enrolment VALUES (112233,
                  'FIT9001',1,2018,80,'HD');
COMMIT;
```

COMMIT makes the changes to the database permanent.

ROLLBACK will undo the changes.

SQL – DML (sequence)

重要程度：⭐⭐⭐⭐
难易程度：⭐⭐

- ❖ 为什么PK最好选择numeric data？
- ❖ auto increment for PK

- Steps
  1. Create sequence
  2. Access sequence(currval & nextval)

– Create sequence

```
CREATE SEQUENCE sno_seq
INCREMENT BY 1;
```

– Access the sequence using two built-in variables (pseudocolumns):
  • NEXTVAL and CURRVAL

```
- INSERT INTO student
  VALUES(sno_seq.nextval,'Bond','James','01-Jan-1994’);

- INSERT INTO enrolment
  VALUES(sno_seq.currval,’FIT9132’,...’);
```

SQL - Query

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

Search Condition

Comparison
<, >, <=, >=, =, !=
Example: display employee_id which the employee salary is higher than 80,000

```
SELECT emp_id
FROM employee
WHERE emp_salary > 80000;
```

Range:
BETWEEN xxx AND xxx
Example: display employee id which the employee salary is between 10000 and 80000

```
SELECT emp_id
FROM employee
WHERE emp_salary BETWEEN 10000 AND 80000;
```

SQL - Query

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

Search Condition (continue)

Set Membership: IN
To test whether the value of expression equals one of a set of values
city IN ('Melbourne','Sydney') 查找city这一列带有'Melbourne' 或 'Sydney'的行

Pattern Match: LIKE
To test whether a string matches a specified pattern
%: represents any sequence of zero or more characters _: represents any single character

example:

```
SELECT emp_id
FROM employees
WHERE emp_lname LIKE 'C%'
```

找Employee last name 以C开头的

other example:
%m: 以m结尾的
%abc%: 在任何位置带有abc的
_ _ C% : C 在第三位的

SQL – Query (NULL)

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

NULL:
IS NULL / IS NOT NULL: to test whether a column has a NULL value

Example:

```
SELECT *
FROM student
WHERE grade IS NULL;
```

SQL – Query (NVL, AS)

重要程度：★★★★★
难易程度：★★★★

❖ **NVL(colname, value replace NULL) - in SELECT statement**

NVL(enrol_mark, 0) 把enrol_mark 为NULL的替换成0 NVL(enrol_grade,'WH')

❖ **Rename: AS**

SELECT stud_id, enrol_mark/10 AS new_mark FROM enrolment;

SELECT stud_id AS "student id" FROM ENROLMENT;

	STU_NBR	NVL(ENROL_MARK,0)	NVL(ENROL_GRADE,'WH')
1	11111111	78	D
2	11111111	0	WH
3	11111111	0	WH
4	11111112	35	N
5	11111112	0	WH
6	11111113	65	C
7	11111113	0	WH
8	11111114	0	WH

SQL – Query (ORDER BY, NULLS LAST/FIRST)

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

❖ ORDER BY

Ascending & descending: 默认asc
可以有多个colum参与sorting

```
SELECT stud_id
FROM enrolment
ORDER BY enrol_mark DESC;
```

❖ NULLS LAST & NULLS FIRST

NULLS LAST: 如果排序遇到null行， null行放在最后

NULLS FIRST: 如果排序遇到null行， null行放在最前

```
SELECT stud_id
FROM enrolment
ORDER BY enrol_mark DESC NULLS LAST;
```

SQL – Query (DISTINCT)

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

❖ **DISTINCT**

如果出现重复的，只选择一行

```
SELECT DISTINCT stud_id
FROM enrolment
WHERE enrol_mark IS NULL;
```

SQL – Query (Join)

重要程度：★★★★★
难易程度：★★★★

SELECT *
FROM table 1 JOIN table 2 ON join condition
ORDER BY attribute;

STUDENT

sno	name
1	alex
2	maria
3	bob

QUALIFICATION

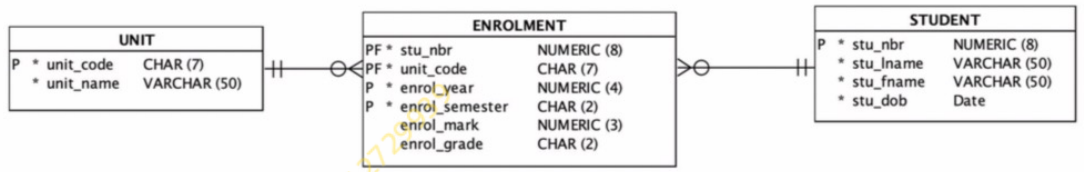
sno	degree	year
1	bachelor	1990
1	master	2000
2	PhD	2001

```
SELECT *
FROM student JOIN qualification
        ON student.sno = qualification.sno
ORDER BY student.sno
```

sno	name	degree	year
1	alex	bachelor	1990
1	alex	master	2000
2	maria	PhD	2001

SQL JOIN

- For database students are **required to use ANSI JOINS**
  - placing the join in the where clause is not acceptable and will be *marked as incorrect for all assessment purposes*
    - such a join is sometimes known as "implicit join notation" - effectively a cross product and then restricted by the where clause
- ANSI JOINS
  - ON
    - the general form which always works, hence the syntax we tend to use
    - FROM student JOIN qualification ON student.sno = qualification.sno
  - USING
    - requires matching attribute names for the PK and FK
    - FROM student JOIN qualification USING (sno)
  - NATURAL
    - requires matching attribute names for the PK and FK
    - FROM student NATURAL JOIN qualification




```
SELECT s.stu_nbr, s.stu_lname, u.unit_name
FROM ((unit u JOIN enrolment e ON u.unit_code=e.unit_code)
        JOIN student s ON e.stu_nbr=s.stu_nbr)
ORDER BY s.stu_nbr, u.unit_name;
```

SQL – Query (CASE…WHEN…THEN...ELSE...END)

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

```
SELECT CASE
   WHEN STUD_FNAME IS NULL THEN STUD_LNAME
   ELSE STUD_LNAME||' '||STUD_FNAME
END AS "STUDENT NAME";
```

Artist Code	Artist Name	Artist State
20	Adrianna Dugget	Victoria
10	Dorette	Victoria
17	Duddy	Victoria
4	Rosalinda Zavattiero	Victoria

SQL – Query (Aggregate Function)

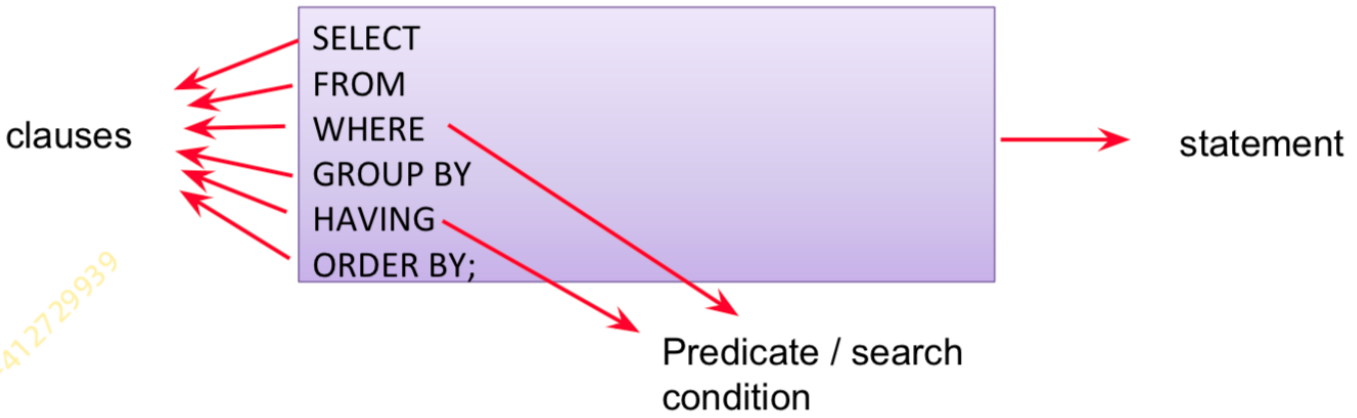
重要程度：★★★★★
难易程度：★★★★

```
SELECT max(mark)
FROM enrolment;
```

```
SELECT min(mark)
FROM enrolment;
```

```
SELECT avg(mark)
FROM enrolment;
```

```
SELECT count(stu_nbr)
FROM enrolment
WHERE mark >= 50;
```



SQL – Query (Aggregate Function)

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

```
SELECT avg(mark)
FROM enrolment;
```

```
SELECT unit_code, avg(mark)
FROM enrolment
GROUP BY unit_code
ORDER BY unit_code;
```

Unit_code	Mark	Studid	Year
FIT2094	80	111	2016
FIT2094	20	111	2015
FIT2004	100	111	2016
FIT2004	40	222	2015
FIT2004	40	333	2015

```
SELECT unit_code, count(*)
FROM enrolment
GROUP BY unit_code
HAVING count(*) > 2;
```

SQL – Query (Aggregate Function)

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

WHERE
- applied to ALL rows

HAVING
- applied to groups

Order:
SELECT
FROM
WHERE
GROUP BY
HAVING

Unit_code	Mark	Studid	Year
FIT2094	80	111	2016
FIT2094	20	111	2015
FIT2004	100	111	2016
FIT2004	40	222	2015
FIT2004	40	333	2015

```
SELECT unit_code, avg(mark), count(*)
FROM enrolmentA
WHERE year = 2015
GROUP BY unit_code
HAVING avg(mark) > 30
ORDER BY avg(mark) DESC;
```

SQL – Query (Subquery)

重要程度：★★★★★
难易程度：★★★★

当需要的条件不在select里面的时候，需要subquery

avg(mark) 不在最后**select**的**output**里，但需要它来对**data**进行比较

"Find all students whose mark is higher than the average mark of all enrolled students"

```
SELECT *
FROM enrolment
WHERE mark > (SELECT avg (mark)
                    FROM enrolment );
```

```
UPDATE enrolment
SET mark = 85
WHERE unit_code = (SELECT unit_code FROM unit WHERE
                          unit_name='Introduction to databases')
        AND mark = 80;
```

SQL – Query (Subquery)

重要程度：★★★★★
难易程度：★★★★

三种类型：

1. **Nested**
   Subquery 独立于外面的query，并且只会执行一次

2. **Correlated**
   每经过一行subquery，就会执行一次，但是里面的query和外部的query有联系，无法独立运行

3. **Inline**
   要通过计算得到的，output经常作为一个‘table’跟在from后面

```
select studid, unitcode, mark
from uni.enrolment
where (unitcode, mark) IN (select unitcode, max(mark)
                           from uni.enrolment
                           group by unitcode)
order by unitcode, studid;
```

check whether.

11111111 FIT1004
11111112 FIT1040
11111112 FIT1004
11111113 FIT1040
11111113 FIT1004

result

UNITCODE	MA
FIT1040	
FIT1004	
FIT5132	
FIT5136	
FIT2077	

excute once.

```
select studid, unitcode, mark from uni.enrolment e1
where mark =
(select max(mark) from uni.enrolment e2
where e1.unitcode = e2.unitcode)
```

unit

```
select studid, e.unitcode, mark
from
 (select unitcode, max(mark) as max_mark
  from uni. enrolment
  group by unitcode) max_table
  join uni.enrolment e on e.unitcode = max_table.unitcode and
                          e.mark = max_table.max_mark
order by unitcode, studid;
```

UNITCODE	MAX_MARK
FIT1040	80
FIT1004	90
FIT5132	78
FIT5136	80
FIT2077	74
FIT5131	88

SQL – Query (View)

重要程度：⭐⭐⭐⭐⭐
难易程度：⭐⭐⭐⭐

虚拟table，一个或多个表依照某个条件组合而成的结果集, 对视图只能进行select操作, 没有实际的物理记录

```
create or replace view max_view as
select unitcode, max(mark) as max_mark
From uni.enrolment
group by unitcode
Select * from max_view
Order by unitcode;
```

SQL – Query (Relational Set Operator)

重要程度：★★★★★
难易程度：★★★★

UnionAll
–All rows selected by either query, including all duplicates ▪ Union (重复也合并)

Union
–All rows selected by either query, removing duplicates (e.g,, DISTINCT on Union All) (去重合并)

Intersect

–All distinct rows selected by both queries (找交集)

Minus

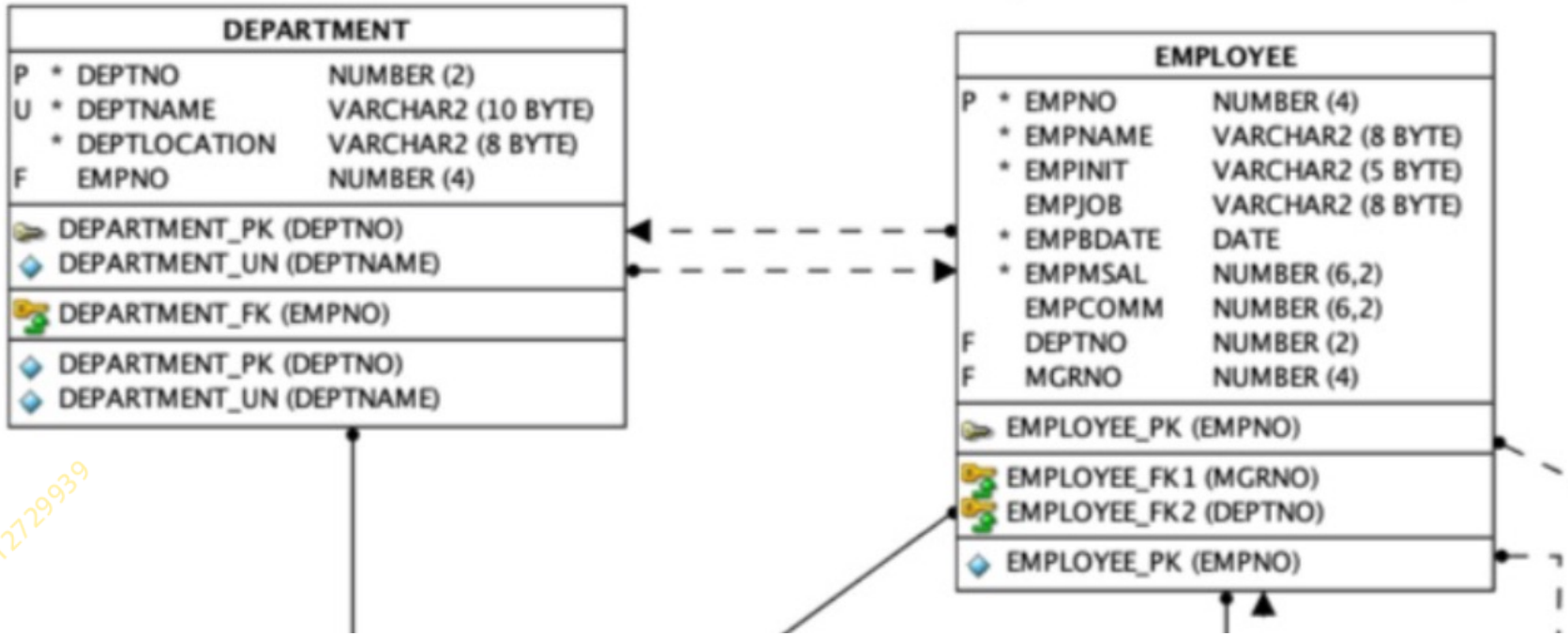
–All distinct rows selected by the first query but not by the second (前表有后表没有的)

例题练习

重要程度：★★★★★
难易程度：★★★★

v. The employee named KING who has a job as the only company DIRECTOR has been assigned to manage the new EXAM department. Record this in the database.

[4 marks]



例题练习

重要程度：
难易程度：

```
UPDATE department
SET
    empno = (
        SELECT
            empno
        FROM
            employee
        WHERE
            empname = 'KING'
            AND empjob = 'DIRECTOR'
    )
WHERE
    deptname = 'EXAM';

COMMIT;
```

重难点总结

重难点总结

SQL QUERY

- where 用于搜索条件，having用于搜索group by之后的条件

- Query顺序为：

```
Select attribute1,attribute2,…
from tablename
where condition (optional)
group by (optional)
having (optional)
order by xxx asc/desc (optional)
```

- 如果所需要的条件不在目前提取的表里时，需要subquery来实现

- 凡是出现了aggregate function，后面都要带group by

- Aggregate function：
- Max(), min(), avg(), sum(), count() …

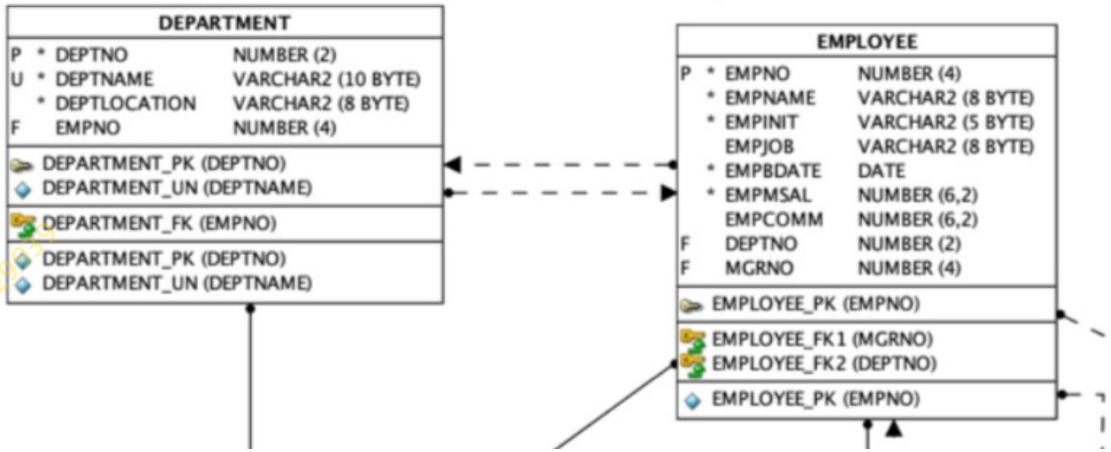
- Join 有left join，right join，full outer join以及join，要根据不同场景选择合适的join

课后作业

iii. List for all employees, the employee number, name, birthdate and the number of *different* courses they have registered for. Note that some employees may repeat a course, this repeat does not count as a different course. Order the output by employee number. Sample output will have the form (only partial shown):

EMPNO	EMPNAME	DOB	CRSCOUNT
7369	SMITH	17-Dec-1982	0
7499	ALLEN	20-Feb-1978	4
7521	WARD	22-Feb-1979	1
7566	JONES	02-Apr-1984	2
7654	MARTIN	28-Sep-1973	0
7698	BLAKE	01-Nov-1980	2

[10 marks]



HD·EDUCATION

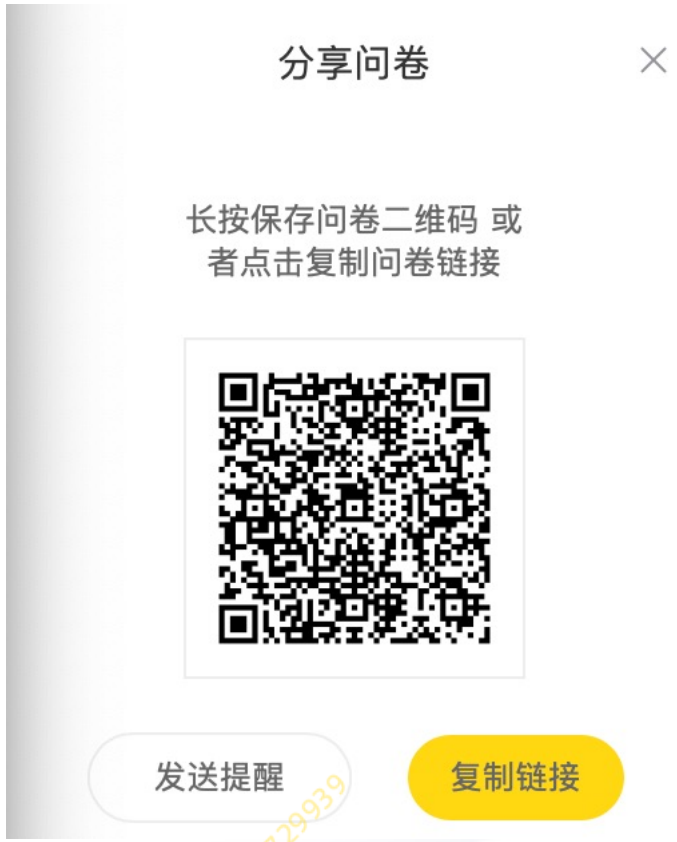
下节课预告

下节课预告

WEEK 11： WEB DATABASE(FIT2094)/NOSQL(FIT9132)

SQL 习题训练讲解

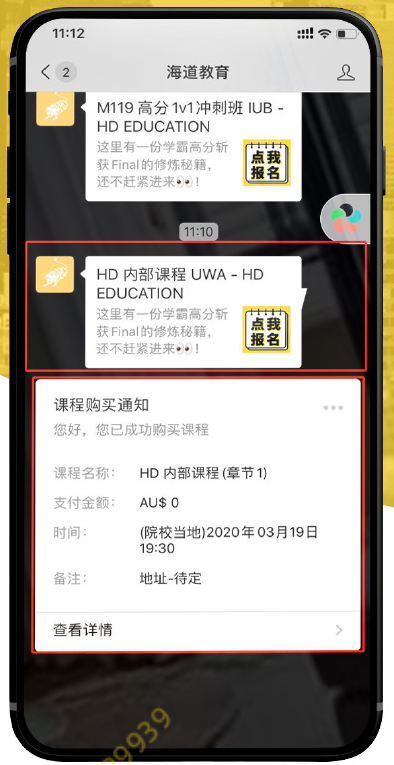
收集反馈



课程结束后，如果您对课程或者服务的任何建议和意见

请给予我们提高和改进的机会，感谢您对 HD·EDUCATION 课程和服务的信任！

· 填写问卷操作流程 ·



第一步

关注【海道教育】服务号后
点击【购买通知】或【上课提醒】

第二步

点击【前往学习】

第三步

点击【去评价】就可以为课程进行评价