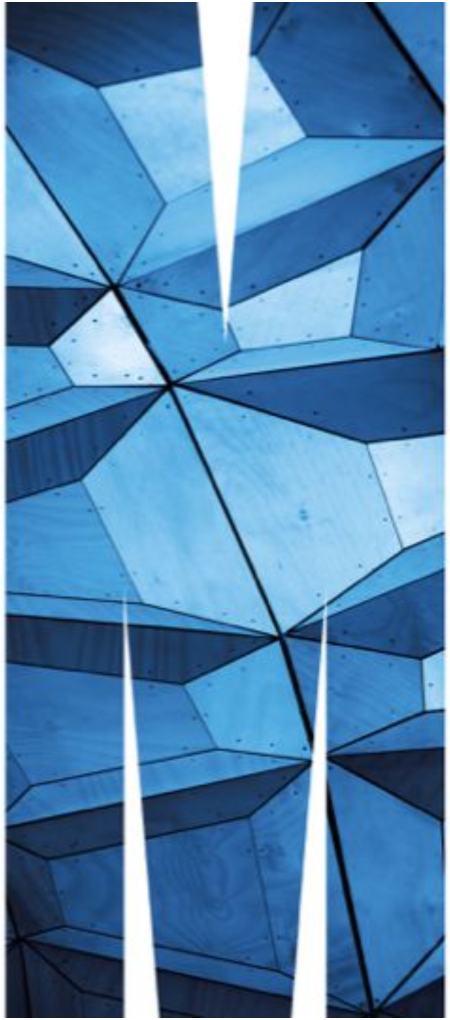
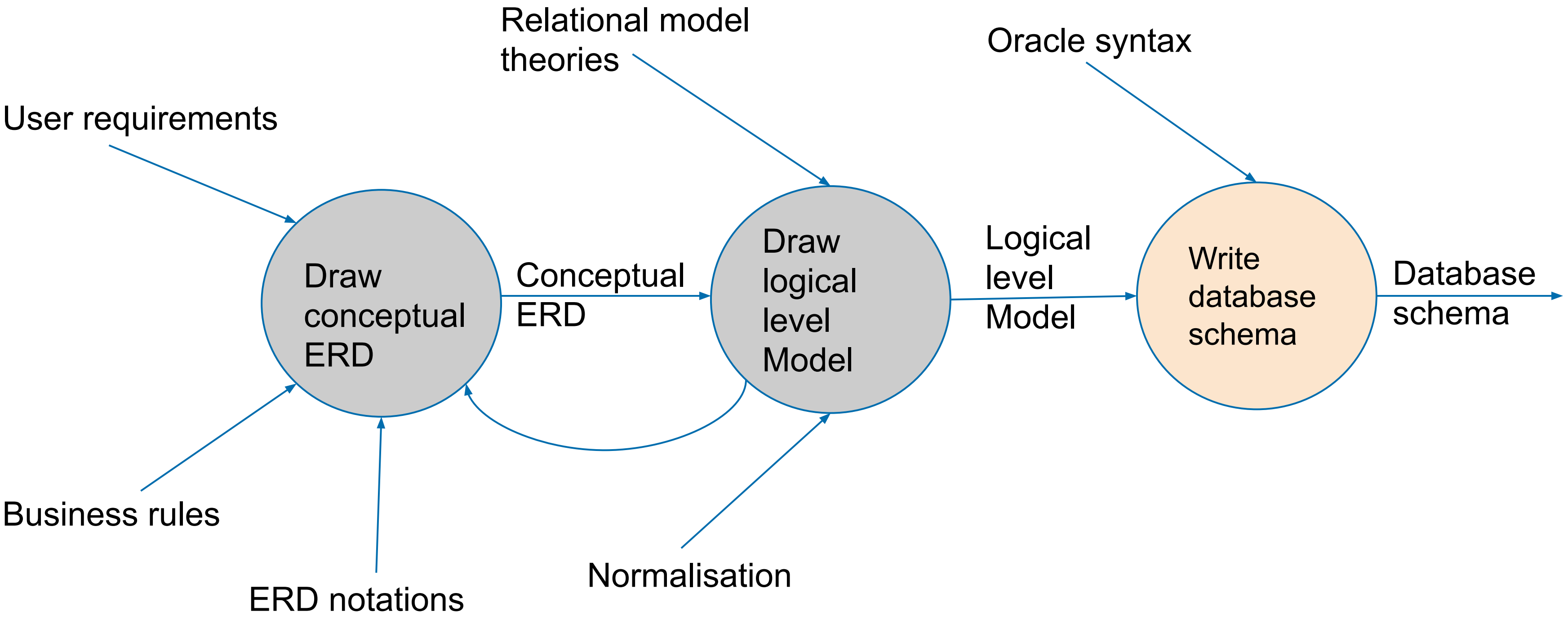
MONASH
INFORMATION
TECHNOLOGY



Creating & Populating the Database

Workshop Q&A 2021S2

Note for this Workshop Q&A we are using examples and syntax particular to Oracle


Relational model theories
Oracle syntax
User requirements
Draw conceptual ERD
Conceptual ERD
Draw logical level Model
Logical level Model
Write database schema
Database schema
Business rules
ERD notations
Normalisation

SQL general syntax

- A single statement is ended with SEMICOLON.
- Predefined KEYWORDs represent clauses (components) of a statement.
- Keywords are NOT case sensitive.
- Examples:

```
CREATE TABLE unit
  (
    unit_code   CHAR(7)NOT NULL,
    unit_name   VARCHAR2(50)CONSTRAINT uq_unit_name UNIQUE NOT NULL,
    CONSTRAINT pk_unit PRIMARY KEY (unit_code)
  );
SELECT * FROM unit;
```

Q1. The SQL Language is made up of the following components (multiple answers can be selected):

A. Data Control Language (DCL)
B. Data Query Language (DQL)
C. Data Definition Language (DDL)
D. Data Manipulation Language (DML)
E. Data Structured Query Language (DSQL)
F. Data Update Language (DUL)

SQL Statements

- Data Definition Language (DDL)
  - Creating database structure
    - CREATE TABLE, ALTER TABLE, DROP TABLE
- Data Manipulation Language (DML)
  - Adding and Manipulating database contents (rows)
    - INSERT, UPDATE, DELETE
  - Retrieving data from database
    - SELECT
- Data Control Language (DCL)
  - Set permissions on objects
    - GRANT

CREATE A TABLE (DDL)

Q2. An attribute is to be used to store Australian postcodes eg. 3001, 2000 - the data type should be:

A. VARCHAR2(4)
B. CHAR(4)
C. NUMBER(4)
D. STRING(4)

Q3. An attributes is to be used to store a customers outstanding balance - the permitted range is $0 to $2000.00 - the data type should be:

A. VARCHAR2(8)
B. NUMBER(4,2)
C. NUMBER(6,2)
D. NUMBER

Q4. An attribute is to be used to store the year an event occurred - the data types which could be used are (multiple answers can be selected):

A. DATE
B. DATE(4)
C. NUMBER(4)
D. CHAR(4)

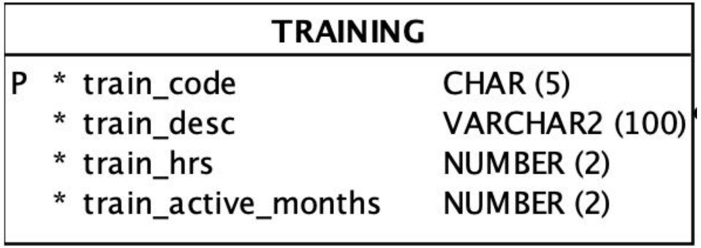
Q5. An attribute is to be used to store the finish time of athletes entered in a 800m running event in minutes and seconds eg. 3 min 25 sec - the data type which should be used is:

A. DATETIME
B. TIME
C. DATE
D. NUMBER(3,2)

Common ORACLE data types

- **Text:** CHAR(size), VARCHAR2(size)
  - e.g., CHAR(10), VARCHAR2(10)
  - CHAR(10) → ‘apple’ = ‘apple   ‘
  - VARCHAR2(10) → ‘apple’ != ‘apple   ‘
- **Numbers:** NUMBER(precision, scale)
  - Weight NUMBER(7) or NUMBER(7,0) → Weight = 7456124
  - Weight NUMBER(9,2) → Weight = 7456123.89
  - Weight NUMBER(8,1) → Weight = 7456123.9
- **Data/Time:** DATE, TIMESTAMP
  - DATE can store a date and time (time to seconds), stored as Julian date
  - TIMESTAMP can store a date and a time (up to fractions of a second)
  - TIMESTAMP WITH TIME ZONE

Column VS Table Level Constraints



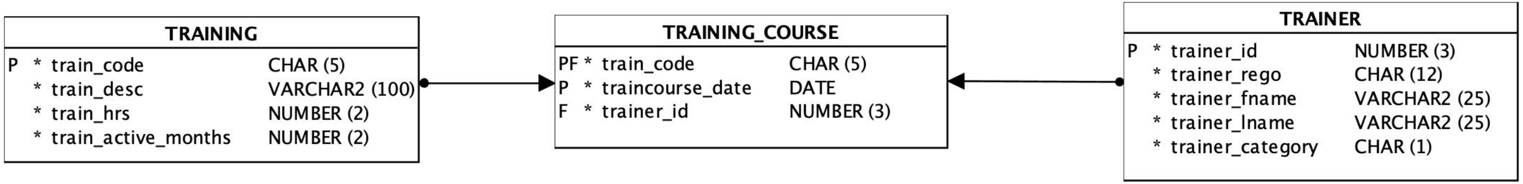
TRAINING

P	*	train_code	CHAR (5)
	*	train_desc	VARCHAR2 (100)
	*	train_hrs	NUMBER (2)
	*	train_active_months	NUMBER (2)

```
CREATE TABLE training (
    train_code              CHAR(5) NOT NULL,
    train_desc              VARCHAR2(100) NOT NULL,
    train_hrs               NUMBER(2) NOT NULL,
    train_active_months  NUMBER(2) NOT NULL,
    CONSTRAINT training_pk PRIMARY KEY (train_code)
);
```

column constraints

table constraint



TRAINING

P	*	train_code	CHAR (5)
	*	train_desc	VARCHAR2 (100)
	*	train_hrs	NUMBER (2)
	*	train_active_months	NUMBER (2)

TRAINING_COURSE

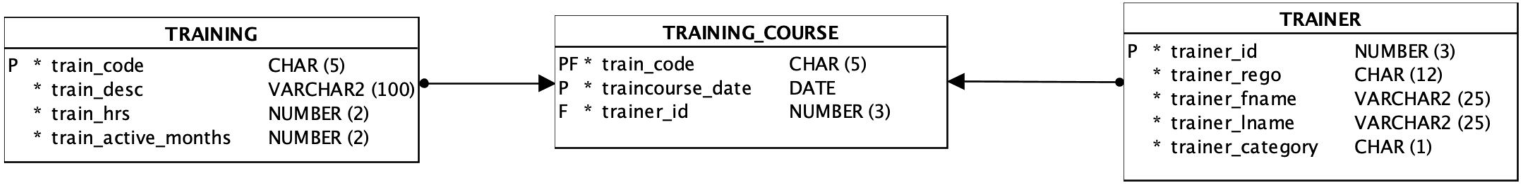
PF	*	train_code	CHAR (5)
P	*	traincourse_date	DATE
F	*	trainer_id	NUMBER (3)

TRAINER

P	*	trainer_id	NUMBER (3)
	*	trainer_rego	CHAR (12)
	*	trainer_fname	VARCHAR2 (25)
	*	trainer_lname	VARCHAR2 (25)
	*	trainer_category	CHAR (1)

```
CREATE TABLE training (
   train_code           CHAR(5) NOT NULL,
   train_desc           VARCHAR2(100) NOT NULL,
   train_hrs            NUMBER(2) NOT NULL,
   train_active_months  NUMBER(2) NOT NULL,
   CONSTRAINT training_pk PRIMARY KEY ( train_code )
);

CREATE TABLE trainer (
   trainer_id        NUMBER(3) NOT NULL,
   trainer_rego      CHAR(12) NOT NULL,
   trainer_fname     VARCHAR2(25) NOT NULL,
   trainer_lname     VARCHAR2(25) NOT NULL,
   trainer_category  CHAR(1) NOT NULL,
   CONSTRAINT trainer_pk PRIMARY KEY ( trainer_id )
);
```



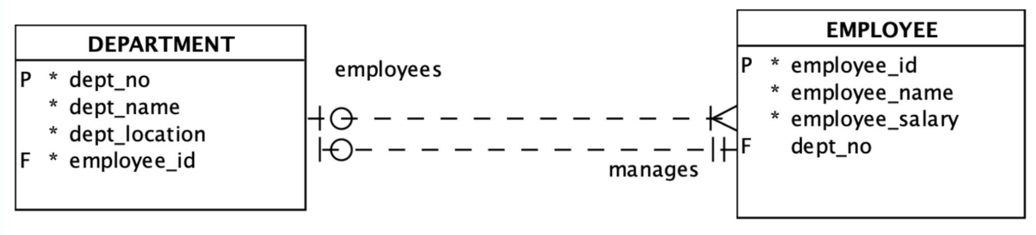
```
TRAINING
P  * train_code            CHAR (5)
   * train_desc            VARCHAR2 (100)
   * train_hrs             NUMBER (2)
   * train_active_months   NUMBER (2)
```

```
TRAINING_COURSE
PF * train_code         CHAR (5)
P  * traincourse_date   DATE
F  * trainer_id         NUMBER (3)
```

```
TRAINER
P  * trainer_id         NUMBER (3)
   * trainer_rego       CHAR (12)
   * trainer_fname      VARCHAR2 (25)
   * trainer_lname      VARCHAR2 (25)
   * trainer_category   CHAR (1)
```

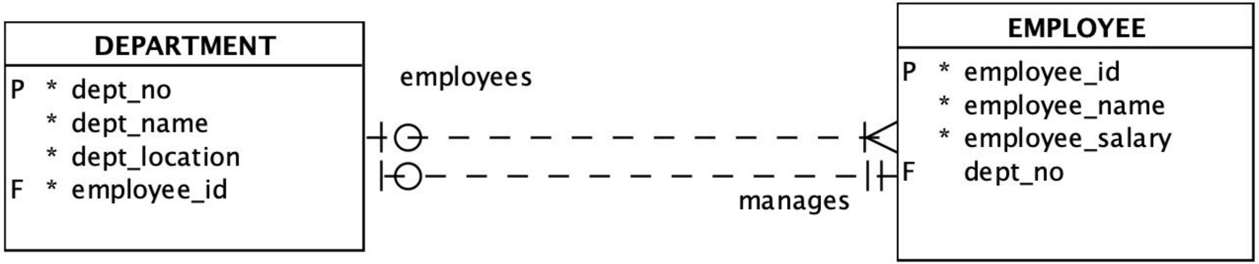
```
CREATE TABLE training_course (
    train_code          CHAR(5) NOT NULL,
    traincourse_date    DATE NOT NULL,
    trainer_id          NUMBER(3) NOT NULL,
    CONSTRAINT training_course_pk PRIMARY KEY ( train_code, traincourse_date ),
    CONSTRAINT trainer_trainingcourse FOREIGN KEY ( trainer_id )
                        REFERENCES trainer ( trainer_id ),
    CONSTRAINT training_trainingcourse FOREIGN KEY ( train_code )
                        REFERENCES training ( train_code )
);
```

Q6. The foreign keys in this model can be declared by (multiple answers can be selected):



A. Column Constraints
B. Table Constraints
C. Use of the ALTER command
D. None of these

Problems here?



Alternative (BETTER) method of defining FKs

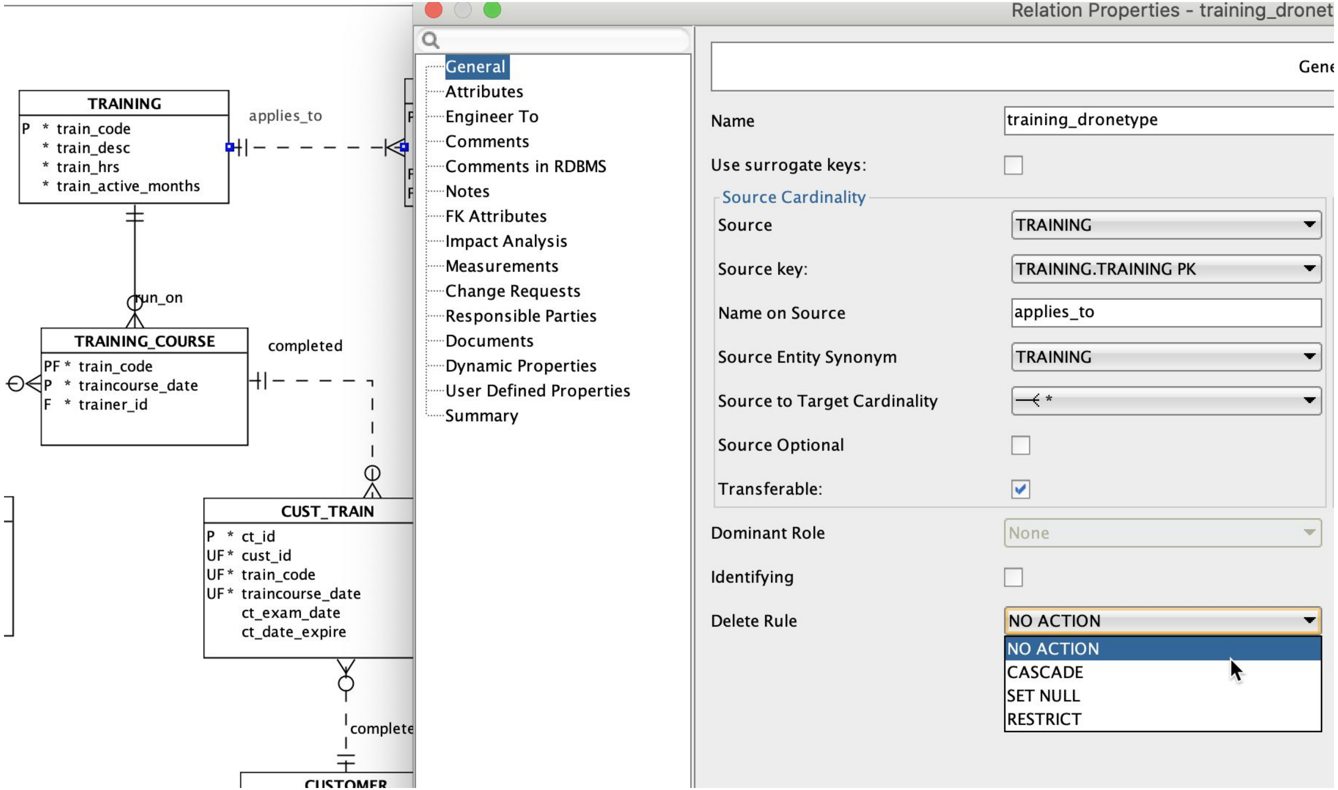
```
CREATE TABLE training_course (
    train_code              CHAR(5) NOT NULL,
    traincourse_date          DATE NOT NULL,
    trainer_id                NUMBER(3) NOT NULL,
    CONSTRAINT training_course_pk PRIMARY KEY ( train_code, traincourse_date )
);

ALTER TABLE training_course
    ADD
            ( CONSTRAINT trainer_trainingcourse FOREIGN KEY ( trainer_id )
                        REFERENCES trainer ( trainer_id ),
             CONSTRAINT training_trainingcourse FOREIGN KEY ( train_code )
                        REFERENCES training ( train_code ));
```

Referential Integrity

- To ensure referential integrity, SQL defines three possible actions for FKs in relations when a deletion of a primary key occurs:
  - RESTRICT (Oracle No Action basically equivalent)
    - Deletion of tuples is NOT ALLOWED for those tuples in the table referred by the FK (the table containing PK) if there is corresponding tuple in the table containing the FK.
  - CASCADE
    - A deletion of a tuple in the table referred by the FK (the table containing PK) will result in the deletion of the corresponding tuples in the table containing the FK.
  - NULLIFY
    - A deletion of a tuple in the table referred by the FK (the table containing PK) will result in the update of the corresponding tuples in the table containing the FK to NULL.

Referential Constraints SQL Data Modeller



Relation Properties - training_dronet

Name: training_dronetype

Use surrogate keys: □

Source Cardinality

Source: TRAINING

Source key: TRAINING.TRAINING PK

Name on Source: applies_to

Source Entity Synonym: TRAINING

Source to Target Cardinality: —< *

Source Optional: □

Transferable: ☑

Dominant Role: None

Identifying: □

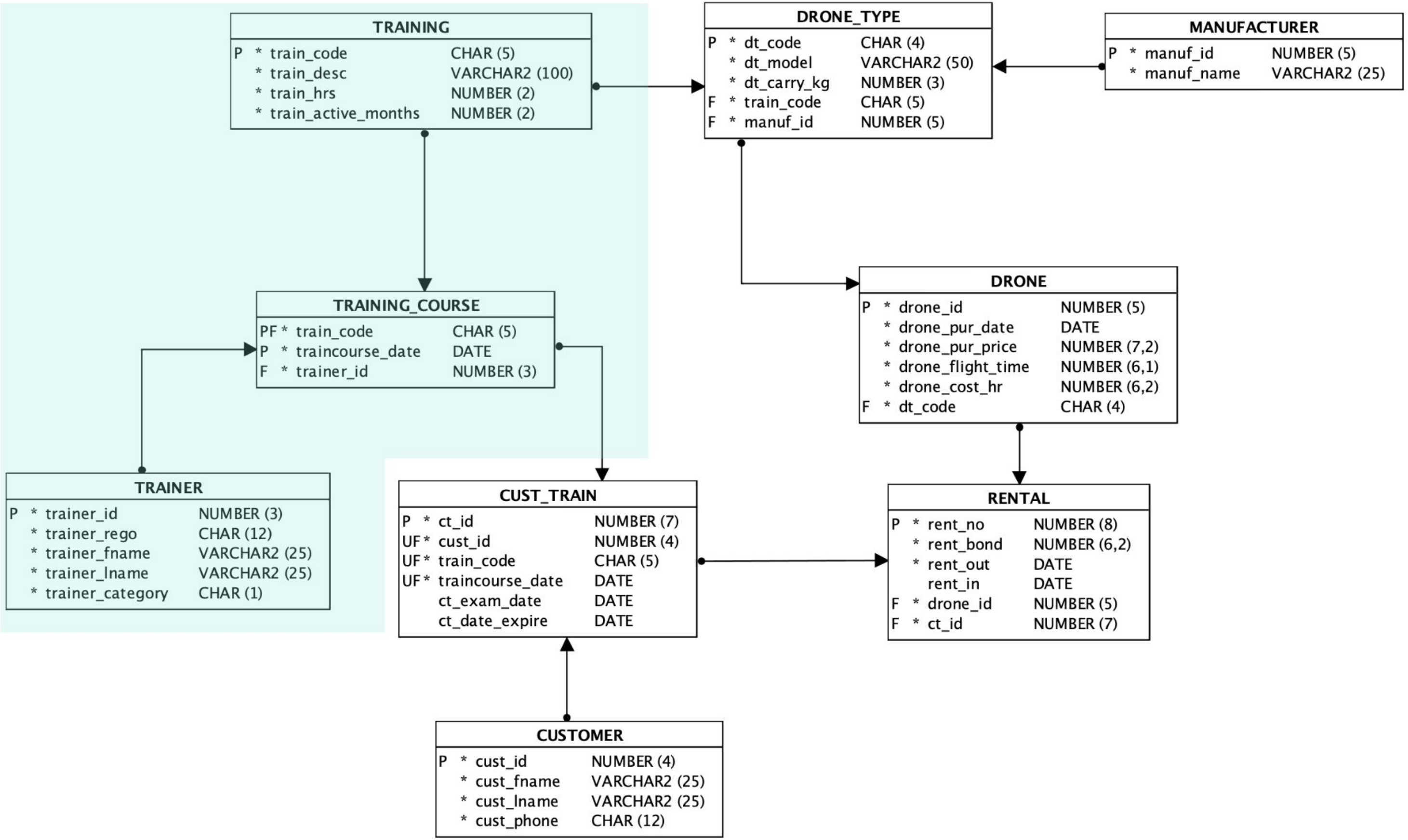
Delete Rule: NO ACTION

- NO ACTION
- CASCADE
- SET NULL
- RESTRICT

What Referential Integrity Constraint to implement?

- Use the model to decide on what referential integrity constraint to implement.
  - Mandatory vs Optional participation.

- **The constraints must be decided at the design phase**.



TRAINING

P	*	train_code	CHAR (5)
	*	train_desc	VARCHAR2 (100)
	*	train_hrs	NUMBER (2)
	*	train_active_months	NUMBER (2)

DRONE_TYPE

P	*	dt_code	CHAR (4)
	*	dt_model	VARCHAR2 (50)
	*	dt_carry_kg	NUMBER (3)
F	*	train_code	CHAR (5)
F	*	manuf_id	NUMBER (5)

MANUFACTURER

P	*	manuf_id	NUMBER (5)
	*	manuf_name	VARCHAR2 (25)

TRAINING_COURSE

PF	*	train_code	CHAR (5)
P	*	traincourse_date	DATE
F	*	trainer_id	NUMBER (3)

DRONE

P	*	drone_id	NUMBER (5)
	*	drone_pur_date	DATE
	*	drone_pur_price	NUMBER (7,2)
	*	drone_flight_time	NUMBER (6,1)
	*	drone_cost_hr	NUMBER (6,2)
F	*	dt_code	CHAR (4)

TRAINER

P	*	trainer_id	NUMBER (3)
	*	trainer_rego	CHAR (12)
	*	trainer_fname	VARCHAR2 (25)
	*	trainer_lname	VARCHAR2 (25)
	*	trainer_category	CHAR (1)

CUST_TRAIN

P	*	ct_id	NUMBER (7)
UF	*	cust_id	NUMBER (4)
UF	*	train_code	CHAR (5)
UF	*	traincourse_date	DATE
		ct_exam_date	DATE
		ct_date_expire	DATE

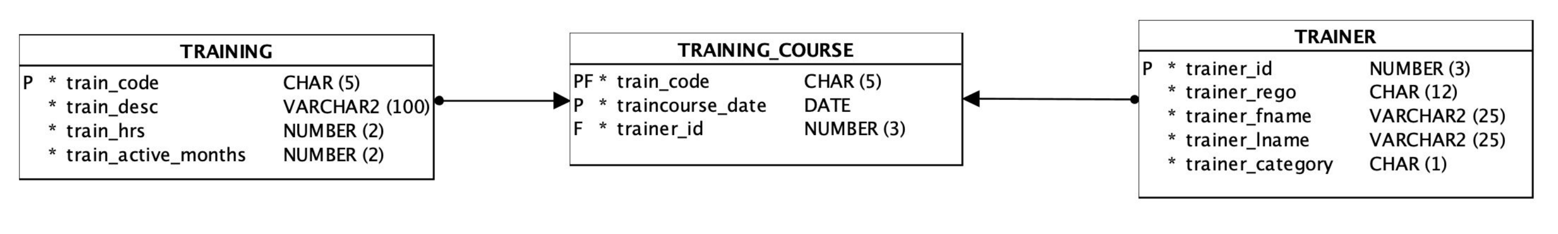
RENTAL

P	*	rent_no	NUMBER (8)
	*	rent_bond	NUMBER (6,2)
	*	rent_out	DATE
		rent_in	DATE
F	*	drone_id	NUMBER (5)
F	*	ct_id	NUMBER (7)

CUSTOMER

P	*	cust_id	NUMBER (4)
	*	cust_fname	VARCHAR2 (25)
	*	cust_lname	VARCHAR2 (25)
	*	cust_phone	CHAR (12)

Q7. TRAINERs have a high turnover (they are employed and leave the company frequently), what could we select for the referential constraint involving TRAINING_COURSE.trainer_id FK (multiple answers can be selected):



TRAINING

P	*	train_code	CHAR (5)
	*	train_desc	VARCHAR2 (100)
	*	train_hrs	NUMBER (2)
	*	train_active_months	NUMBER (2)

TRAINING_COURSE

PF	*	train_code	CHAR (5)
P	*	traincourse_date	DATE
F	*	trainer_id	NUMBER (3)

TRAINER

P	*	trainer_id	NUMBER (3)
	*	trainer_rego	CHAR (12)
	*	trainer_fname	VARCHAR2 (25)
	*	trainer_lname	VARCHAR2 (25)
	*	trainer_category	CHAR (1)

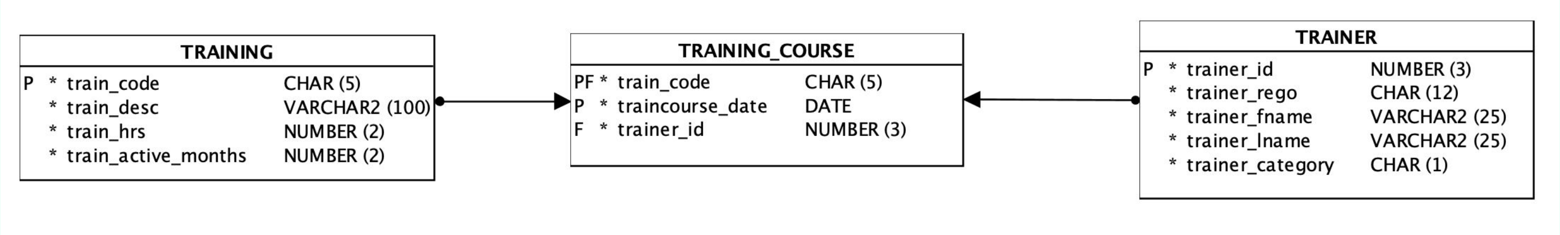
A. RESTRICT

B. SET NULL

C. CASCADE

D. None of these

Q8. What could we select for the referential constraint involving TRAINING.train_code FK (multiple answers can be selected):



TRAINING

P	*	train_code	CHAR (5)
	*	train_desc	VARCHAR2 (100)
	*	train_hrs	NUMBER (2)
	*	train_active_months	NUMBER (2)

TRAINING_COURSE

PF	*	train_code	CHAR (5)
P	*	traincourse_date	DATE
F	*	trainer_id	NUMBER (3)

TRAINER

P	*	trainer_id	NUMBER (3)
	*	trainer_rego	CHAR (12)
	*	trainer_fname	VARCHAR2 (25)
	*	trainer_lname	VARCHAR2 (25)
	*	trainer_category	CHAR (1)

A. RESTRICT
B. SET NULL
C. CASCADE
D. None of these

ALTER TABLE

- Used to change a tables structure.
- For example:
  - Adding column(s).
  - Removing column(s).
  - Adding constraint(s) - used previously for FK's, but can be any constraint
  - Removing constraint(s)

```
ALTER TABLE TRAINER
  ADD (CONSTRAINT chk_trainercategory CHECK
             (trainer_category IN ( 'C', 'F' )),
        trainer_nocourses number(3) DEFAULT 0 NOT NULL);
```

Manipulate Constraints

- Turn constraint ON or OFF to temporarily disable
  - ALTER TABLE training_course
    DISABLE CONSTRAINT training_trainingcourse;
  - ALTER TABLE training_course
    ENABLE CONSTRAINT training_trainingcourse;

- Remove/re add constraint to modify constraint
  - ALTER TABLE training_course
    DROP CONSTRAINT training_trainingcourse;
  - ALTER TABLE training_course
    ADD
    (CONSTRAINT training_trainingcourse FOREIGN KEY (train_code)
    REFERENCES training (train_code) **ON DELETE CASCADE**);

DELETING A TABLE

- Use the DROP statement.
- Examples:
  - `DROP TABLE training_course PURGE;`
  - `DROP TABLE trainer` **`CASCADE CONSTRAINTS PURGE`**`;`

ADDING TUPLES/ROWS TO A TABLE (DML)

INSERT

- Adding data to a table in a database.
- SYNTAX:

  INSERT INTO *table [(column [, column...])]*

  VALUES *(value [, value...]);*

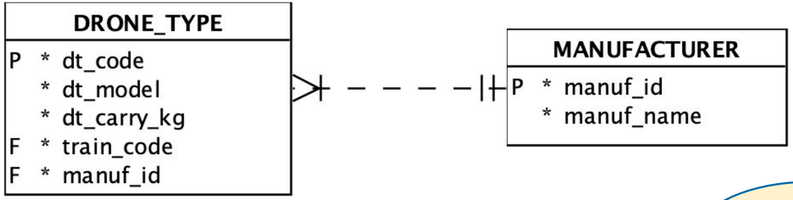
```
INSERT INTO training VALUES ('COOO1','Starter Drone Training 1',8,24);
```

```
INSERT INTO trainer (trainer_id, trainer_rego, trainer_fname, trainer_lname,
trainer_category) VALUES (312,'DR523412-314','Thomas','Price','F');
```

```
INSERT INTO training_course VALUES ('COOO1','20-Oct-2020',312);
```

Role of: to_date and to_char

COMMIT and ROLLBACK



DRONE_TYPE

P * dt_code
* dt_model
* dt_carry_kg
F * train_code
F * manuf_id

MANUFACTURER

P * manuf_id
* manuf_name

How to determine

```
INSERT INTO manufacturer VALUES (12, 'DJI');

INSERT INTO drone_type VALUES('DJIT', 'DJI Trello', 5, 'C0001', 12);
```

COMMIT makes the changes to the database permanent.
ROLLBACK will undo the changes.

COMMIT/ROLLBACK **only applicable to INSERT/UPDATE and DELETE**

Using a SEQUENCE

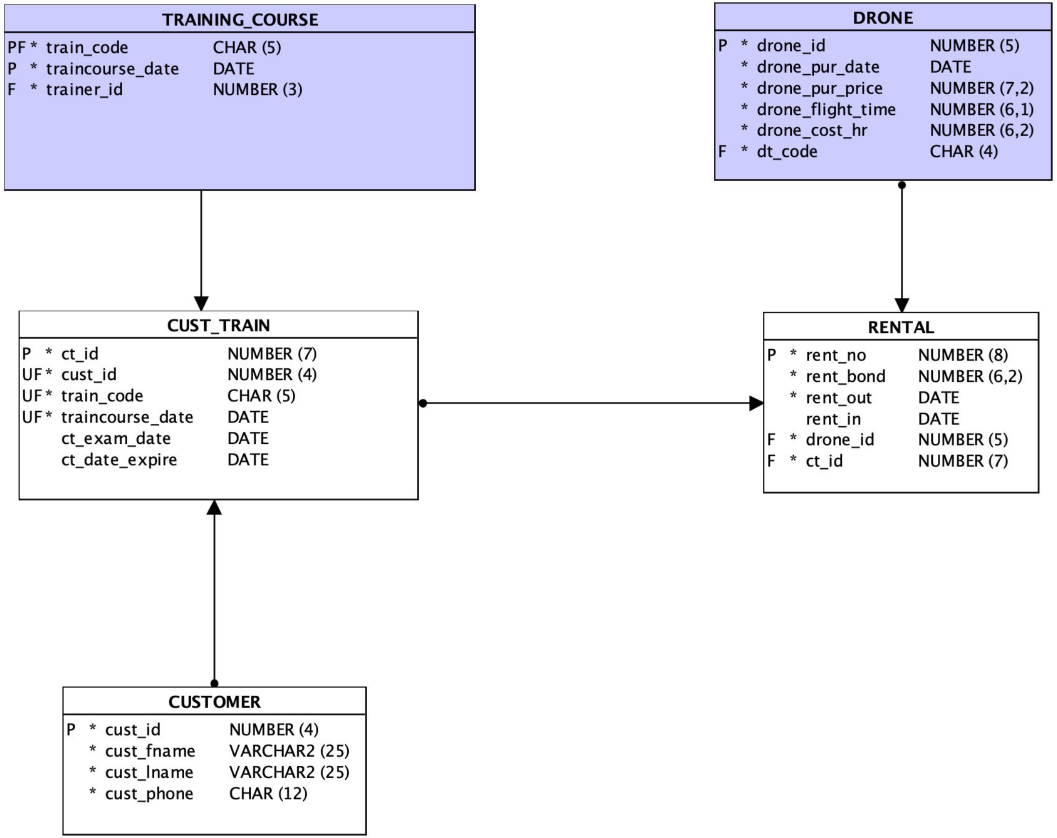
- Oracle supports auto-increment of a numeric PRIMARY KEY.
  - SEQUENCE.
- Steps to use:
  - Create sequence

    ```
    CREATE SEQUENCE manuf_seq
    INCREMENT BY 1;
    ```

  - Access the sequence using two built-in variables (pseudocolumns):
    - NEXTVAL and CURRVAL
      - `INSERT INTO manufacturer VALUES(manuf_seq.nextval,'DJI');`

      - `INSERT INTO drone_type VALUES('DJIT', 'DJI Trello', 5, 'C0001', manuf_seq.currval);`
  - **Note sequence value CANNOT be relied on after a COMMIT/ROLLBACK**

PUTTING THIS TO WORK



TRAINING_COURSE

PF	*	train_code	CHAR (5)
P	*	traincourse_date	DATE
F	*	trainer_id	NUMBER (3)

DRONE

P	*	drone_id	NUMBER (5)
	*	drone_pur_date	DATE
	*	drone_pur_price	NUMBER (7,2)
	*	drone_flight_time	NUMBER (6,1)
	*	drone_cost_hr	NUMBER (6,2)
F	*	dt_code	CHAR (4)

CUST_TRAIN

P	*	ct_id	NUMBER (7)
UF	*	cust_id	NUMBER (4)
UF	*	train_code	CHAR (5)
UF	*	traincourse_date	DATE
		ct_exam_date	DATE
		ct_date_expire	DATE

RENTAL

P	*	rent_no	NUMBER (8)
	*	rent_bond	NUMBER (6,2)
	*	rent_out	DATE
		rent_in	DATE
F	*	drone_id	NUMBER (5)
F	*	ct_id	NUMBER (7)

CUSTOMER

P	*	cust_id	NUMBER (4)
	*	cust_fname	VARCHAR2 (25)
	*	cust_lname	VARCHAR2 (25)
	*	cust_phone	CHAR (12)

Assume purple relations (tables) have been created

```
CREATE TABLE customer (
    cust_id     NUMBER(4) NOT NULL,
    cust_fname  VARCHAR2(25) NOT NULL,
    cust_lname  VARCHAR2(25) NOT NULL,
    cust_phone  CHAR(12) NOT NULL
);


CREATE TABLE rental (
    rent_no  NUMBER(8) NOT NULL,
    rent_bond  NUMBER(6, 2) NOT NULL,
    rent_out   DATE NOT NULL,
    rent_in    DATE,
    drone_id   NUMBER(5) NOT NULL,
    ct_id      NUMBER(7) NOT NULL
);


CREATE TABLE cust_train (
    ct_id             NUMBER(7) NOT NULL,
    cust_id           NUMBER(4) NOT NULL,
    train_code        CHAR(5) NOT NULL,
    traincourse_date  DATE NOT NULL,
    ct_exam_date      DATE,
    ct_date_expire    DATE
);
```

```
-- PKs
ALTER TABLE rental ADD CONSTRAINT rental_pk PRIMARY KEY ( rent_no );

ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY ( cust_id );

ALTER TABLE cust_train ADD CONSTRAINT cust_train_pk PRIMARY KEY ( ct_id );

-- FKs
ALTER TABLE cust_train
    ADD CONSTRAINT customer_custtrain FOREIGN KEY ( cust_id )
        REFERENCES customer ( cust_id );

ALTER TABLE cust_train
    ADD CONSTRAINT traincourse_custtrain FOREIGN KEY ( train_code,
                                                       traincourse_date )
        REFERENCES training_course ( train_code,
                                     traincourse_date );

ALTER TABLE rental
    ADD CONSTRAINT drone_rental FOREIGN KEY ( drone_id )
        REFERENCES drone ( drone_id );

ALTER TABLE rental
    ADD CONSTRAINT custtrain_rental FOREIGN KEY ( ct_id )
        REFERENCES cust_train ( ct_id );
```

```
-- Other Constraints

ALTER TABLE cust_train
    ADD CONSTRAINT cust_train_uq UNIQUE ( cust_id,
                                          train_code,
                                          traincourse_date );
```

During the workshop Q&A column comments were omitted to speed up the coding.
****HOWEVER** column comments MUST always be provided with the final schema**