| | |
|---|---|
| NAME: | Madhumitha K |
| ROLL NO: | 241801145 |
| DEPT: | AI&DS |

3. Create the animals table. Write the syntax you will use to create the table.

```
CREATE TABLE animals (
    animal_id          NUMBER(6)    CONSTRAINT pk_animals PRIMARY KEY,
    name               VARCHAR2(25),
    license_tag_number NUMBER(10)   CONSTRAINT unq_license UNIQUE,
    admit_date         DATE         CONSTRAINT nn_admit NOT NULL,
    adoption_id        NUMBER(5),
    vaccination_date   DATE         CONSTRAINT nn_vaccine NOT NULL
);
```

Results | Explain | Describe | Saved SQL | History

```
Table created.
```

4. Enter one row into the table. Execute a SELECT * statement to verify your input. Refer to the graphic
below for input.
ANIMAL_I D NAM E LICENSE_TAG_NUMBE R ADMIT_DAT E ADOPTION_I D VACCINATION_DAT E
101 Spot 35540 10-Oct-2004 205 12-Oct-2004

```
SELECT * FROM animals;
```

Results | Explain | Describe | Saved SQL | History

| ANIMAL_ID | NAME | LICENSE_TAG_NUMBER | ADMIT_DATE | ADOPTION_ID | VACCINATION_DATE |
|---|---|---|---|---|---|
| 101 | Spot | 35540 | 10/10/2004 | 205 | 10/12/2004 |

5. Write the syntax to create a foreign key (adoption_id) in the animals table that has a corresponding
primary-key reference in the adoptions table. Show both the column-level and table-level syntax. Note
that because you have not actually created an adoptions table, no adoption_id primary key exists, so the
foreign key cannot be added to the animals table.

```sql
SELECT constraint_name, constraint_type, table_name
FROM user_constraints
WHERE table_name IN ('ADOPTIONS', 'ANIMALS');
```

Results | Explain | Describe | Saved SQL | History

| CONSTRAINT_NAME | CONSTRAINT_TYPE | TABLE_NAME |
|---|---|---|
| PK_ADOPTIONS | P | ADOPTIONS |
| PK_ANIMALS | P | ANIMALS |
| FK_ADOPT | R | ANIMALS |