Schemas:

1. **Student**
   - Attributes: `ssn`, `name`, `address`, `major`
2. **Course**
   - Attributes: `code`, `title`
3. **Registered**
   - Attributes: `ssn`, `code`

Questions:

1. List the codes of courses in which at least one student is registered (registered courses).

2. List the titles of registered courses (of those in 1.).

3. Show the student’s details with CSE major.

4. List the codes of courses for which no student is registered.

5. Students who are not registered to any courses.

5.1. Students' names who are not registered to any courses.

6. The titles of courses for which no student is registered.

7. Names of students and the titles of courses they registered to.

8. Names of students who are registered for ‘Database Systems’ or ‘Analysis and Design’.

9. SSNs of students who are registered for both ‘Database Systems’ and ‘Analysis of Algorithms’.

10. The name of those students who are registered for both ‘Database Systems’ and ‘Analysis of Algorithms’.

Answers:

1. List the codes of courses in which at least one student is registered (registered courses).

Relational Algebra Expression:

```
∏ code (Registered)
```

---

2. List the titles of registered courses (of those in 1.).

Relational Algebra Expression:

```
∏ title (Course ⋈ Registered)
```

---

3. Show the student’s details with CSE major.

Relational Algebra Expression:

```
σ major = "CSE" (Student)
```

---

4. List the codes of courses for which no student is registered.

Relational Algebra Expression:

```
∏ code (Course) - ∏ code (Registered)
```

---

5. Students who are not registered to any courses.

Relational Algebra Expression:

```
∏ ssn (Student) - ∏ ssn (Registered)
```

5.1. Students' names who are not registered to any courses.

Relational Algebra Expression (Option 1):

```
∏ name (Student) - ∏ name (Student ⋈ Registered)
```

Relational Algebra Expression (Option 2):

```
∏ name ((∏ ssn (Student) - ∏ ssn (Registered)) ⋈ Student)
```

---

6. The titles of courses for which no student is registered.

Relational Algebra Expression (Option 1):

```
∏ title ((∏ code (Course) - ∏ code (Registered)) ⋈ Course)
```

Relational Algebra Expression (Option 2):

```
∏ title (Course) - ∏ title (Registered ⋈ Course)
```

---

7. Names of students and the titles of courses they registered to.

Relational Algebra Expression:

```
∏ s.name, c.title (Student ⋈ Registered ⋈ Course)
```

---

8. Names of students who are registered for ‘Database Systems’ or ‘Analysis and Design’.

Relational Algebra Expression:

```
∏ name (σ title = "Database Systems" ∨ title = "Analysis and Design" (Student ⋈
Registered ⋈ Course))
```

---

9. SSNs of students who are registered for both ‘Database Systems’ and ‘Analysis of Algorithms’.

Relational Algebra Expression:

```
(∏ ssn (σ title = "Database Systems" (Registered ⋈ Course)))
∩
(∏ ssn (σ title = "Analysis of Algorithms" (Registered ⋈ Course)))
```

---

10. The name of those students who are registered for both ‘Database Systems’ and ‘Analysis of Algorithms’.

Relational Algebra Expression:

```
(∏ name (σ title = "Database Systems" (Student ⋈ Registered ⋈ Course)))
∩
(∏ name (σ title = "Analysis of Algorithms" (Student ⋈ Registered ⋈ Course)))
```

Explanation of Symbols Used:

1. **σ (Selection):** Filters rows based on a condition.
2. **Π (Projection):** Selects specific columns (attributes) from a relation.
3. ∪ **(Union):** Combines two relations, removing duplicates.
4. **∩ (Intersection):** Finds common elements between two relations.
5. **− (Difference):** Finds elements in one relation that are not in another.
6. ⋈ **(Natural Join):** Combines two relations based on common attributes.
7. **× (Cartesian Product):** Produces all possible combinations of tuples from two relations.

This completes the schemas, questions, and answers!