Relational Database Schema:

$\mathcal{R}$ is a set of relational schemes $\{R_1, \ldots, R_n\}$.

A relational scheme $R$ consists of a set of attributes $\{A_1, \ldots, A_k\}$. Each attribute $A_i$ is associated with a domain denoted as $\text{dom}(A_i)$.

A relation, $r$ of $R$ is a finite set of mappings $\{t_1, \ldots, t_n\}$ from $R$ to $\text{dom}(A_1) \times \ldots \times \text{dom}(A_k)$.

Use $t[A_i]$ to denote the value $A_i$ of $t$. An instance $I$ of $\mathcal{R}$ is a mapping from $\mathcal{R}$ to relations.

$I(R_i) = r_i$ where $r_i$ is the relation of $R_i$, e.g., $R = \{R_1, R_2\}$

Consider the relational database schema:

\[
\begin{align*}
\mathcal{R} &= \{R_1, R_2\} \\
R_1 &= \{A_1, \ldots, A_k\} \\
R_2 &= \{B_1, \ldots, B_l\} \\
r_1 &= \{t_1, \ldots, t_p\} \\
r_2 &= \{s_1, \ldots, s_q\} \\
t_1 &= \{A_1 \rightarrow 1, A_2 \rightarrow 2, \ldots, A_k \rightarrow k\} \\
s_1 &= \{B_1 \rightarrow l, B_2 \rightarrow l-1, \ldots, B_1 \rightarrow 1\} \\
I &= \{R_1 \rightarrow r_1, R_2 \rightarrow r_2\} \\
I(R_1) &= r_1 \\
t_1[A_2] &= 2
\end{align*}
\]

Relational Operators:

Select : $\sigma_c$

$\sigma_c(r) = \{t \in r \mid t \text{ satisfies the given select condition} \}$

Project : $\Pi_{A_1, \ldots, A_n}$

$\Pi_A(r) = \{t[A] \mid t \in r\}$

Cartesian Product : $\times$

$r \times s = \{t_1, t_2 \mid t_1 \in r, t_2 \in s\}$

Union : $\cup$

$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$

These four operators form what is called as the SPCU algebra.
Difference : 
\[ r - s = \{ t | t \in r \text{ and } t \not\in s \} \]

Intersect : \( \cap \)
This can be derived using the following:
\[
\begin{align*}
    r \cap s &= r - (r - s) \\
    &= s - (s - r)
\end{align*}
\]

Therefore the intersect operator is not considered to be a basic operator.

Properties about queries or Query Languages:

Containment: \( Q_1 \subseteq Q_2 \) if for every input database \( D \), \( Q_1(D) \subseteq Q_2(D) \). That is the result of applying \( Q_1 \) on \( D \) is contained in the result of applying \( Q_2 \) on \( D \).

Equivalence: \( Q_1 = Q_2 \) if \( Q_1 \subseteq Q_2 \) and \( Q_2 \subseteq Q_1 \)

Satisfiability: A query \( Q \) is satisfiable if for some database \( D \) \( Q(D) \) is non-empty. This means that when you run a query \( Q \) on the database \( D \) some output is achieved.

Monotonicity: A query \( Q \) is monotonic if for every database \( D_1 \) and \( D_2 \), \( D_1 \subseteq D_2 \), implies \( Q(D_1) \subseteq Q(D_2) \).

Rule-Based Conjunctive Queries:

Let \( \mathcal{R} \) be a relational database schema. A RBCQ over \( \mathcal{R} \) is an expression of the form:
\[
\begin{align*}
    \text{Ans}(y_1, \ldots, y_k) :&= R_1(x'_1, x'_2, \ldots, x'_{m_1}), \ldots, R_k(x'_1, x'_2, \ldots, x'_{m_k}) \\
    \text{Ans}(\overline{y}) :&= R_1(\overline{x}), \ldots, R_k(\overline{x})
\end{align*}
\]
where \( \overline{x} \) are vectors of variables and \( k \geq 0 \)

The LHS is called the head and the RHS is called the body of the expression.

\[
\begin{align*}
    \{y_1, \ldots, y_r\} &\subseteq \{x'_1, \ldots, x'_{m}, \ldots, x'_1, \ldots, x'_{m_k}\} \\
    \overline{y} &\subseteq \overline{x} \cup \ldots \cup \overline{x_k}
\end{align*}
\]

Semantics of such queries:

Consider: \( \text{Emp}(\text{ssn}, \text{name}, \text{deptid}), \text{Dept}(\text{id}, \text{budget}) \)
\( \text{Ans}(n) \leftarrow \text{Emp}(s, n, d), \text{Dept}(d, b) \)
Find \( n \), such that \( s, d, b \) from \( \text{Emp}, \text{Dept} \).

\[
\{ n | \exists s, \exists d, \exists b, (\text{Emp}(s, n, d) \land \text{Dept}(d, b)) \}
\]
Semantics:
Let $Q$ denote the query:
$$\text{Ans}(\overline{x}) : -R_1(\overline{x_1}), \ldots, R_n(\overline{x_n})$$
over the database schema $R$ and let $D$ denote an instance over $R$,
$$Q(D) = \{ \mu(\overline{x}) | \mu \text{ is a valuation over } \text{var}(Q) \text{ and } \mu(\overline{x_i}) \in D(R_i) \text{ for each } i \in [i,n] \}$$

$Q(D)$ is the image of $D$ under $Q$.
$\text{var}(Q)$ denotes the set of variables in $Q$.
A valuation is a mapping from variables to constants.

In the previous example
$$\mu_1 = \{ s \rightarrow 123, n \rightarrow \text{Mary}, d \rightarrow 01, b \rightarrow 20K \}$$
$$\mu_2 = \{ s \rightarrow 124, n \rightarrow \text{Joe}, d \rightarrow 01, b \rightarrow 30K \}$$

Proofs:

Fact 1: Every RBCQ is satisfiable
$\text{Ans}(n) : \text{Emp}(s,n,d), \text{Dept}(d,b)$
Define a canonical instance of $Q$

$Q^c$:

$$Q(Q^c) = \{ \text{Ans}(n^c) \}$$

Proof:- Construct a canonical instance of $Q$. Pick a different constant for each variable:

$$\mu : \text{var}(Q) \xrightarrow{1-1} \text{dom}$$

$$Q^c = \mu(Q)$$

Hence, $Q(Q^c)$ is non empty because $\text{Ans}(\mu(\overline{x})) \in Q(Q^c)$