

5) Eliminiamo i duplicati:

Vediamo che la query così come è ci ritorna tutti i risultati che vogliamo, anche troppi!!!

Troppe colonne “data”, righe duplicate... non ci siamo ancora ma quasi!

Prendiamo come base la prima delle tre query, e vediamo di non far prendere alla seconda query le righe che di sicuro sono presenti nella prima. In pratica dobbiamo prendere solo quei risultati che non hanno un riscontro nella prima query. Per farlo possiamo modificare la seconda query così:

```
SELECT chiuso.data AS chiuso_data, chiusi, numero_errato.data AS numero_errato_data, errati,  
inviata_mail.data as inviata_mail_data, mail FROM chiuso  
RIGHT OUTER JOIN  
numero_errato  
ON chiuso.data = numero_errato.data  
LEFT OUTER JOIN  
inviata_mail  
ON numero_errato.data = inviata_mail.data  
WHERE chiuso.data IS NULL  
ORDER BY numero_errato.data
```

Lo stesso ragionamento va applicato alla terza query: prendiamo solo i risultati che non hanno corrispondenza né nella prima né nella seconda query:

```
SELECT chiuso.data AS chiuso_data, chiusi, numero_errato.data AS numero_errato_data, errati,  
inviata_mail.data AS inviata_mail_data, mail  
FROM chiuso  
RIGHT OUTER JOIN  
(numero_errato  
RIGHT OUTER JOIN  
inviata_mail  
ON numero_errato.data = inviata_mail.data)  
ON chiuso.data = inviata_mail.data  
WHERE chiuso.data IS NULL AND numero_errato.data IS NULL  
ORDER BY inviata_mail.data
```

Il risultato dell'unione di queste tre query così modificate è questo:

chiuso_data	chiusi	numero_errato_data	errati	inviata_mail_data	mail
2009-08-02	1	NULL	NULL	NULL	NULL
2009-08-05	1	2009-08-05	1	NULL	NULL
2009-08-06	1	NULL	NULL	2009-08-06	1
2009-08-08	1	2009-08-08	1	2009-08-08	1
NULL	NULL	2009-08-03	1	NULL	NULL
NULL	NULL	2009-08-07	1	2009-08-07	1
NULL	NULL	NULL	NULL	2009-08-04	1

Ancora qualche ritocchino per evitare le colonne data duplicate e per mettere al posto dei NULL degli “zeri”. Nel frattempo ordiniamo anche il risultato finale eliminando gli ordinamenti parziali delle singole query (servivano solo a facilitare la comprensione dei risultati intermedi)

Nelle query selezioniamo solo il campo data che siamo sicuri sia presente, quindi nella prima query chiuso.data, nella seconda numero_errato.data e nella terza inviata_mail.data.

```
(SELECT chiuso.data,  
        COALESCE(chiusi,0) AS chiusi,  
        COALESCE(errati,0) AS errati,  
        COALESCE(mail,0) AS mail  
FROM chiuso  
LEFT OUTER JOIN  
numero_errato  
ON chiuso.data = numero_errato.data  
LEFT OUTER JOIN  
inviata_mail  
ON chiuso.data = inviata_mail.data)  
UNION ALL  
(SELECT numero_errato.data,  
        COALESCE(chiusi,0) AS chiusi,  
        COALESCE(errati,0) AS errati,  
        COALESCE(mail,0) AS mail  
FROM chiuso  
RIGHT OUTER JOIN  
numero_errato  
ON chiuso.data = numero_errato.data  
LEFT OUTER JOIN  
inviata_mail  
ON numero_errato.data = inviata_mail.data  
WHERE chiuso.data IS NULL)  
UNION ALL  
(SELECT inviata_mail.data,  
        COALESCE(chiusi,0) AS chiusi,  
        COALESCE(errati,0) AS errati,  
        COALESCE(mail,0) AS mail  
FROM chiuso  
RIGHT OUTER JOIN  
(numero_errato  
RIGHT OUTER JOIN  
inviata_mail  
ON numero_errato.data = inviata_mail.data)  
ON chiuso.data = inviata_mail.data  
WHERE chiuso.data IS NULL AND numero_errato.data IS NULL)  
ORDER BY data
```

data	chiusi	errati	mail
2009-08-02	1	0	0
2009-08-03	0	1	0
2009-08-04	0	0	1
2009-08-05	1	1	0
2009-08-06	1	0	1
2009-08-07	0	1	1
2009-08-08	1	1	1

P.S. Anche se ho ricontrollato il tutto, qualche refuso è sempre possibile che sia presente. Se c'è qualcosa che non torna, avvisatemi!!! Di seguito c'è lo sviluppo della query senza l'uso delle viste.

```
(SELECT chiuso.data,
        COALESCE(chiusi,0) AS chiusi,
        COALESCE(errati,0) AS errati,
        COALESCE(mail,0) AS mail
FROM (SELECT data, COUNT(*) AS chiusi
FROM utenti_ebay
WHERE stato = 'Chiuso'
GROUP BY data) AS chiuso
LEFT OUTER JOIN
(SELECT data, COUNT(*) AS errati
FROM utenti_ebay
WHERE stato = 'Numero errato'
GROUP BY data) AS numero_errato
ON chiuso.data = numero_errato.data
LEFT OUTER JOIN
(SELECT data, COUNT(*) AS mail
FROM utenti_ebay
WHERE stato = 'Inviata mail'
GROUP BY data) AS inviata_mail
ON chiuso.data = inviata_mail.data)
UNION ALL
(SELECT numero_errato.data,
        COALESCE(chiusi,0) AS chiusi,
        COALESCE(errati,0) AS errati,
        COALESCE(mail,0) AS mail
FROM (SELECT data, COUNT(*) AS chiusi
FROM utenti_ebay
WHERE stato = 'Chiuso'
GROUP BY data) AS chiuso
RIGHT OUTER JOIN
(SELECT data, COUNT(*) AS errati
FROM utenti_ebay
WHERE stato = 'Numero errato'
GROUP BY data) AS numero_errato
ON chiuso.data = numero_errato.data
LEFT OUTER JOIN
(SELECT data, COUNT(*) AS mail
FROM utenti_ebay
WHERE stato = 'Inviata mail'
GROUP BY data) AS inviata_mail
ON numero_errato.data = inviata_mail.data
WHERE chiuso.data IS NULL)
UNION ALL
(SELECT inviata_mail.data,
        COALESCE(chiusi,0) AS chiusi,
        COALESCE(errati,0) AS errati,
        COALESCE(mail,0) AS mail
FROM (SELECT data, COUNT(*) AS chiusi
FROM utenti_ebay
WHERE stato = 'Chiuso'
GROUP BY data) AS chiuso
RIGHT OUTER JOIN
((SELECT data, COUNT(*) AS errati
FROM utenti_ebay
WHERE stato = 'Numero errato'
GROUP BY data) AS numero_errato
RIGHT OUTER JOIN
(SELECT data, COUNT(*) AS mail
FROM utenti_ebay
WHERE stato = 'Inviata mail'
GROUP BY data) AS inviata_mail
ON numero_errato.data = inviata_mail.data)
ON chiuso.data = inviata_mail.data
WHERE chiuso.data IS NULL AND numero_errato.data IS NULL)
ORDER BY data
```